République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara – Boumerdès



Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques Informatique
Filière : Informatique
Spécialité : SI

N° de l'Arrêté d'habilitation : arrêté n°872 du 26/07/2016

Mémoire de Fin d'Études
En vue de l'obtention du diplôme de Licence Académique

Assistance au Pré-diagnostic Radiologique par Recherche d'Images Similaires

---

Présenté par :
Chiba Yasmine
Rahim Naima
Zemmour Malak
Boukeloua Imane

Soutenu le .. / 06 / 2026 devant le jury composé de :

Dr. TOUAZI FAYCAL	:	Examinateur
Pr. GACEB Djamel	:	Encadreur

Dédicace

À nos parents, pour leur amour et leur sacrifice infinis, qui ont illuminé notre parcours universitaire.

À nos encadreurs, pour leur patience, leur rigueur scientifique et leur bienveillance constante.

À nos amis et valeureux camarades de promotion.

Remerciements

Nous souhaitons avant tout exprimer notre profonde gratitude à ALLAH, le Tout-Puissant, pour nous avoir donné la force, le courage et la patience nécessaires à l'accomplissement de ce modeste travail.

Nous adressons nos plus chaleureux remerciements à notre encadreur M. Gaceb Djamel pour ses précieux conseils et son soutien indéfectible tout au long de ce projet. Un grand merci se dirige à Mr. Touazi pour tous ses efforts déployés pour évaluer ce travail.

Résumé

Dans le cadre de la transformation numérique de la santé, l'analyse automatique des imageries thoraciques constitue un enjeu clinique majeur pour le dépistage précoce des pathologies pulmonaires. Face aux limites des systèmes de classification traditionnels, qui se heurtent souvent au manque de données médicales annotées et à la rigidité des prédictions strictes, ce projet propose une approche alternative basée sur le paradigme de la recherche d'images par le contenu (CBIR) optimisée par apprentissage métrique profond.

L'objectif principal de ce travail est de concevoir et de réaliser un système intelligent d'aide au pré-diagnostic radiologique capable de projeter des clichés de radiographies pulmonaires au sein d'un espace latent hautement discriminant. En exploitant l'architecture des réseaux siamois (Siamese Networks), le système apprend une fonction de similarité sémantique capable de rapprocher les profils pathologiques analogues et éloigner les cas dissimilaires. Lors de la soumission d'une nouvelle radiographie requête, le moteur de recherche extrait de manière transparente les cas historiques les plus similaires cliniquement validés, offrant ainsi aux praticiens un historique comparatif fiable pour sécuriser leur prise de décision. Cette solution, implémentée sous l'écosystème Python et PyTorch, démontre la pertinence des approches par comparaison pour pallier le coût élevé des expertises d'annotation en imagerie médicale.

Abstract

In the context of digital health transformation, the automated analysis of chest X-rays represents a critical clinical challenge for early detection of pulmonary diseases. To overcome the limitations of conventional classification systems, which frequently suffer from a lack of annotated medical datasets and rigid output structures, this project introduces an alternative approach based on the Content-Based Image Retrieval (CBIR) paradigm powered by deep metric learning.

The core objective of this study is to design and implement an intelligent diagnostic assistant capable of projecting chest radiographs into a highly discriminative latent embedding space. By leveraging twin Siamese Networks, the system effectively learns a semantic similarity metric that clusters analogous pathological profiles while isolating normal cases. When presented with a query X-ray, the retrieval engine extracts the most clinically relevant historical cases, providing radiologists with a solid comparative baseline to secure their final diagnosis. Implemented using Python and PyTorch, this framework demonstrates the high efficiency of similarity-based learning in mitigating the scarcity and high cost of expert annotations in medical imaging.

Table des matières

Chapitre 1 : Définitions et étude bibliographique sur l'apprentissage profond, l'approche CBIR et les réseaux siamois 10
1.1 Introduction 10
1.2 Notions de base sur l'intelligence artificielle et le deep learning 10
1.2.1 De l'IA au Deep Learning 10
1.2.2 Réseaux de neurones convolutifs (CNN) 10
1.2.2.1 DenseNet121 12
1.2.2.2 TorchXRayVision DenseNet121 12
1.2.2.3 EfficientNet-B0 12
1.2.2.4 ResNet-50 12
1.3 Apprentissage par transfert (Transfer Learning) 12
a) 12
1.4 Approche CBIR en imagerie médicale 12
1.5 Réseaux siamois 13
1.5.1 Fonctions de perte 14
1.6 L'apprentissage métrique 14
1.7 Conclusion 15
Chapitre 2 : Approches élaborées pour la recherche d'images radiologiques similaires **16**
2.1 Introduction 16
2.2 Méthodologie 16
2.3 Prétraitement des données 16
2.4 Indexation des vecteurs de caractéristique avec FAISS 17
2.5 Approches proposées 17
2.5.1 Approche 1 : Niveaux de fine-tuning sur le modèle DenseNet121 pré-entraîné sur ImageNet 17
2.5.1.1 Scénario 1 : Sans fine-tuning (mode extracteur de caractérstique) 17
2.5.1.2 Scénario 2 : mode fine-tuning 17
2.5.2 Approche 2 : Comparaison des fonctions de perte avec TorchXRayVision DenseNet121 et Visual Hard Mining 17
2.5.2.1 Scénario 1 : Contrastive Loss 17
2.5.2.2 Scénario 2 : Triplet loss 18
2.5.3 Approche 3 : Configuration avec différents backbones selon la meilleure configuration des aproches précédentes 18
2.5.3.1 Scénario 1 : TorchXRayVision DenseNet121 18
2.5.3.2 Scénario 2 : EfficientNet-B0 18
2.5.3.3 Scénario 3 : ResNet-50 19
2.6 Conclusion 19
Chapitre 3 : Expérimentations, résultats et discussion **20**
3.1 Introduction 20
3.2 Environnement de développement 20
3.2.1 Environnement matériel 20
3.2.2 Outils et langage de développement 20
3.3 Description de la base de données utilisée 21
3.4 Métriques d'évaluation 22

3.5 Expérimentations et résultats : 22
3.5.1 Expérimentation 1 (Approche 1) 22
3.5.1.1 Scénario 1 : le modèle DenseNet121 Sans fine-tuning 22
3.5.1.2 Scénario 2 : le modèle DenseNet121 Avec fine-tuning 22
3.5.2 Expérimentation 2 (Approche 2) 23
3.5.2.1 Scénario 1 : TorchXRayVision DenseNet121 avec Contrastive Loss et Visual Hard Mining 23
3.5.2.2 Scénario 2 : TorchXRayVision DenseNet121 avec Triplet Loss et Visual Hard Mining 24
3.5.3 Expérimentation 3 (Approche 3) 24
3.5.3.1 Scénario1: TorchXRayVision DenseNet121 24
3.5.3.2 Scénario2 : EfficientNet B0 25
3.5.3.3 Scénario3 : ResNet-50 25
3.6 Discussion générale des résultats 25
3.7 Analyse visuelle avec Grad-Cam 25
3.8 Interface graphique 26
3.8.1 Vue générale de l'interface graphique de l'application 26
3.8.2 Le développement de l'application 26
3.9 Conclusion 27
4 Conclusion générale 28
5 Bibliographie 29

Table des figures

Figure 1 :Architecture standard d'un modèle CNN. 12
Figure 2 :Mécanisme de rétropropagation du gradient. 12
Figure 3 :Architecture fonctionnelle globale d'un système CBIR. 13
Figure 4 :Application du CBIR aux radiographies thoraciques. 13
Figure 5 :Architecture fonctionnelle d'un réseau de neurones siamois. 14
Figure 6 :Exemples de radiographies pulmonaires avant et aprés OpenCV. 16
Figure 7 :Architecture de l'expérimentation 1 Scénario2. ..... **Erreur ! Signet non défini.**
Figure 8 : Architecture de l'expérimentation 2 Scénario1. 18
Figure 9 : Architecture de l'expérimentation 2 Scénario2. 18
Figure 10 :Architecture de l'expérimentation 3 Scénario2. 19
Figure 11 :Architecture de l'expérimentation 3 Scénario3. 19
Figure 12 : Logo de Python 20
Figure 13 :Logo de PyTorch. 20
Figure 14 :Logo de Kaggle 20
Figure 15 :Logo de Flask 21
Figure 16 :Logo OpenCv 21
Figure 17 :Logo Pycharm 21
Figure 18 :Echantilon d'image de dataset 21
Figure 19 :Courbes d'apprentissage du scénario 2 de l'expérimentation 1. 22
Figure 20 : Courbes d'apprentissage du scénario 2 de l'expérimentation 1. 23
Figure 21 : Courbes d'apprentissage du scénario 1 de l'expérimentation 2. 23
Figure 22 : Courbes d'apprentissage du scénario 2 de l'expérimentation 2. 24
Figure 23 : Courbes d'apprentissage du scénario 1 de l'expérimentation 3. 24
Figure 24 :Courbes d'apprentissage du scénario 2 de l'expérimentation 3. 25
Figure 25 : Courbes d'apprentissage du scénario 3 de l'expérimentation 3. 25
Figure 26 :page d'accueil de l 'application RadioSim 26
Figure 27 :page de sélection du modèle 26
Figure 28 :page de traitement de l'image 27
Figure 29 :page de resultats avec Grad-cam 27

=

Liste des tableaux

Tableau 1 : Comparatif des architectures de réseaux CNN. 12
Tableau 2 :Comparatif des métriques d'évaluation sur l'ensemble de test pour l'expérimentation 1 23
Tableau 3 : Comparatif des métriques d'évaluation sur l'ensemble de test pour l'expérimentation 2. 24
Tableau 4 :Comparatif des métriques d'évaluation sur l'ensemble de test pour l'expérimentation 3. 25

Liste des sigles et abréviations

IA : Intelligence Artificielle
ML : Machine Learning (Apprentissage Automatique)
DL : Deep Learning (Apprentissage Profond)
ANN : Artificial Neural Network (Réseau de Neurones Artificiels)
CNN : Convolutional Neural Network (Réseau de Neurones Convolutif)
CBIR : Content-Based Image Retrieval (Recherche d'Images par le Contenu)
CLAHE : Contrast Limited Adaptive Histogram Equalization (Égalisation Adaptative d'Histogramme à Contraste Limité)
GPU : Graphics Processing Unit (Processeur Graphique)
CPU : Central Processing Unit (Processeur Central)
API : Application Programming Interface (Interface de Programmation Applicative)
P@K : Precision at K (Précision parmi les K premiers résultats)
R@K : Recall at K (Rappel parmi les K premiers résultats)
mAP@K : Mean Average Precision at K (Moyenne des Précisions Moyennes à K)
nDCG@K : Normalized Discounted Cumulative Gain at K (Gain Cumulé Actualisé Normalisé à K)
MRR@5 : Mean Reciprocal Rank at 5 (Rang Réciproque Moyen à 5)

Introduction Générale

L'imagerie médicale joue aujourd'hui un rôle essentiel dans le diagnostic et le suivi des maladies respiratoires. À l'échelle mondiale, la pneumonie reste l'une des principales causes de mortalité, touchant particulièrement les enfants de moins de cinq ans et les personnes âgées. En Algérie, les infections respiratoires aiguës constituent un défi de santé publique permanent, accentuant la charge sur les services d'imagerie et la nécessité d'un pré-diagnostic rapide et précis pour guider la prise en charge thérapeutique.

Parmi les examens les plus utilisés, la radiographie pulmonaire constitue un outil rapide, accessible et largement adopté pour la détection de plusieurs pathologies pulmonaires telles que la pneumonie, la tuberculose ou encore la COVID-19. Cependant, son interprétation nécessite une expertise médicale importante et peut parfois être difficile, notamment dans les régions souffrant d'un manque de radiologues spécialisés.

Avec l'évolution du Deep Learning, plusieurs solutions intelligentes ont été développées pour assister les médecins dans l'analyse des images médicales. Les CNN ont démontré des performances remarquables dans ce domaine, mais nécessitent un grand volume de données annotées, ce qui représente une difficulté majeure en milieu médical en raison des contraintes de confidentialité, du coût élevé de l'annotation et de la rareté de certaines pathologies.

Face à ces limitations, les réseaux siamois (Siamese Networks) ont émergé comme une alternative prometteuse. Contrairement aux modèles classiques, ils apprennent à mesurer la similarité entre deux images plutôt qu'à les classifier, ce qui les rend particulièrement adaptés aux contextes où les données d'entraînement sont limitées.

Le volume croissant des examens radiologiques pulmonaires, combiné à la pénurie de radiologues spécialisés dans de nombreuses régions, génère un risque réel d'erreurs diagnostiques et de retards de prise en charge. Les systèmes de classification automatique par apprentissage profond, bien que performants, se heurtent à une contrainte fondamentale dans le domaine médical : la rareté des données annotées. Étiqueter des images cliniques exige une expertise médicale coûteuse, et certaines pathologies demeurent peu représentées dans les bases existantes. Cette réalité rend les modèles supervisés classiques fragiles, peu généralisables et difficiles à déployer en conditions réelles.

Dans ce contexte, notre mémoire propose la conception d'un système intelligent de recherche d'images médicales similaires basé sur les réseaux siamois et les techniques CBIR (Content-Based Image Retrieval). Le système permettra, lorsqu'un utilisateur fournit une radiographie pulmonaire, de rechercher automatiquement les n cas les plus similaires présents dans une base de données médicale, accompagnés de leurs diagnostics confirmés. Cette approche par similarité contourne l'obstacle de l'annotation massive en apprenant directement une métrique de proximité clinique, offrant ainsi aux praticiens un outil d'aide à la décision fiable et adaptable.

Ce mémoire se compose de 3 chapitres organisés comme suit : le premier chapitre est consacré à l'étude bibliographique des concepts théoriques liés au Deep Learning, au CBIR et aux réseaux siamois. Le deuxième présente l'approche proposée et la conception globale du système. Le troisième chapitre détaille l'implémentation, les expérimentations menées et l'analyse des résultats obtenus. Le mémoire s'achève par une conclusion générale et des perspectives de travaux futurs.

1 Chapitre 1 : Définitions et étude bibliographique sur l'apprentissage profond, l'approche CBIR et les réseaux siamois

1.1 Introduction

La radiographie pulmonaire est l'un des examens d'imagerie médicale les plus pratiqués dans le monde. Avec l'augmentation constante du volume de clichés à analyser et la pénurie de radiologues dans certaines régions, le recours à des outils d'aide automatisée au diagnostic devient une nécessité clinique réelle. Dans ce contexte, le Deep Learning offre des capacités d'extraction automatique de caractéristiques visuelles qui surpassent les méthodes classiques, tandis que l'apprentissage par transfert (transfer learning) permet de pallier la rareté des données médicales annotées.

Ce chapitre pose les bases théoriques de notre système : nous y présentons la hiérarchie de l'intelligence artificielle, le paradigme du CBIR appliqué à l'imagerie médicale, le fonctionnement des réseaux siamois et de l'apprentissage métrique, ainsi que les métriques d'évaluation utilisées pour juger la qualité d'un moteur de recherche d'images.

1.2 Notions de base sur l'intelligence artificielle et le deep learning

1.2.1 De l'IA au Deep Learning

L'intelligence artificielle (IA), désigne un ensemble de techniques informatiques permettant à une machine d'imiter certaines capacités humaines, comme le raisonnement, la perception, la reconnaissance de formes ou la prise de décision. Elle regroupe plusieurs approches, allant des systèmes basés sur des règles définies manuellement jusqu'aux systèmes capables d'apprendre automatiquement à partir des données.

Le Machine Learning, ou apprentissage automatique, est une branche de l'intelligence artificielle. Il consiste à entraîner un modèle à partir de données afin qu'il puisse reconnaître des relations, faire des prédictions ou prendre des décisions sans être explicitement programmé pour chaque situation. On distingue généralement trois grands types d'apprentissage : l'apprentissage supervisé, qui utilise des données étiquetées ; l'apprentissage non supervisé, qui cherche à découvrir des structures cachées dans les données ; et l'apprentissage par renforcement, où un agent apprend à agir en fonction de récompenses reçues.

Le Deep Learning, ou apprentissage profond, est une sous-branche du Machine Learning basée sur les réseaux de neurones artificiels comportant plusieurs couches. Ces réseaux apprennent progressivement des représentations de plus en plus complexes à partir des données. Par exemple, dans le cas des images, les premières couches peuvent détecter des éléments simples comme les contours ou les formes, tandis que les couches plus profondes apprennent des caractéristiques plus abstraites. Cette capacité à apprendre automatiquement les caractéristiques rend le Deep Learning particulièrement efficace dans les domaines de la vision par ordinateur, du traitement du langage naturel et de l'analyse de données complexes. [REF ?]

1.2.2 Réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs, ou CNN, sont des architectures de Deep Learning largement utilisées pour le traitement et l'analyse des images. Ils sont conçus pour extraire automatiquement des caractéristiques visuelles à partir des données, comme les contours, les formes, les textures ou les objets.

Le principe principal d'un CNN repose sur l'opération de convolution. Cette opération consiste à appliquer un filtre, appelé noyau, sur différentes zones de l'image afin de produire des cartes de caractéristiques. Grâce au partage des poids, le même filtre est utilisé sur toute l'image, ce qui permet de réduire le nombre de paramètres par rapport à un réseau entièrement connecté classique.

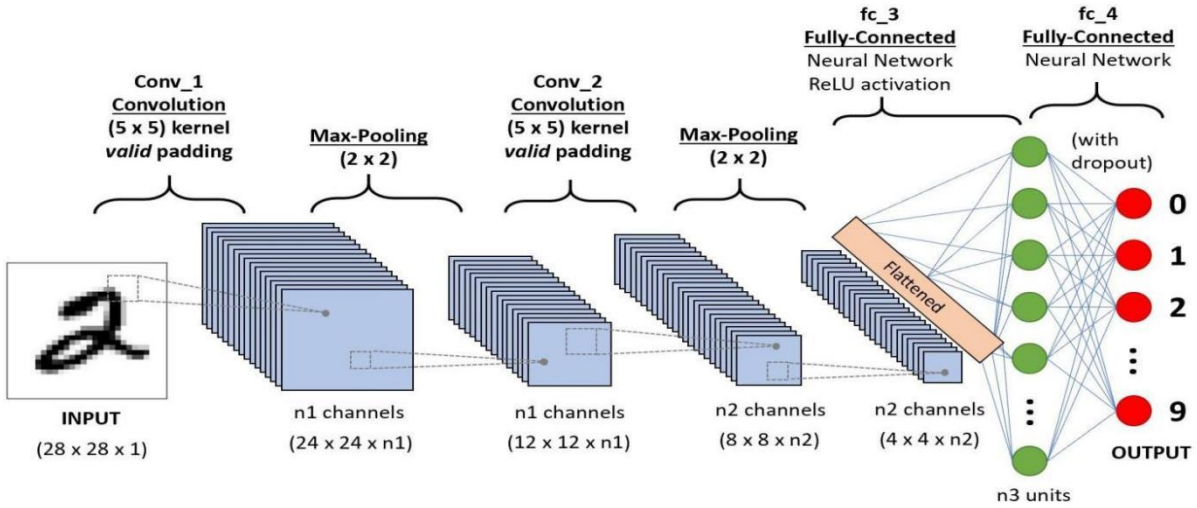

Figure 1:Architecture standard d'un modèle CNN.

Un CNN est généralement composé de plusieurs types de couches. Les couches de convolution permettent d'extraire les caractéristiques visuelles, les couches de pooling réduisent la taille des cartes de caractéristiques tout en conservant les informations importantes, et les couches entièrement connectées sont souvent utilisées à la fin du réseau pour produire une décision de classification ou une représentation finale. Les deux premiers types de couches composent la partie convolutive qui se charge de la descrption automatique de l'image d'entrée, alors que le troisième types de couches forme la seconde partie qui se charge de la classification.

Des fonctions d'activation non linéaires, comme ReLU, sont utilisées entre les couches afin de permettre au modèle d'apprendre des relations complexes dans les données. L'entraînement du réseau se fait généralement par rétropropagation du gradient, où le modèle ajuste progressivement ses poids durant l'apprentissage afin de réduire l'erreur entre la sortie prédite et la sortie attendue [Ref ?].

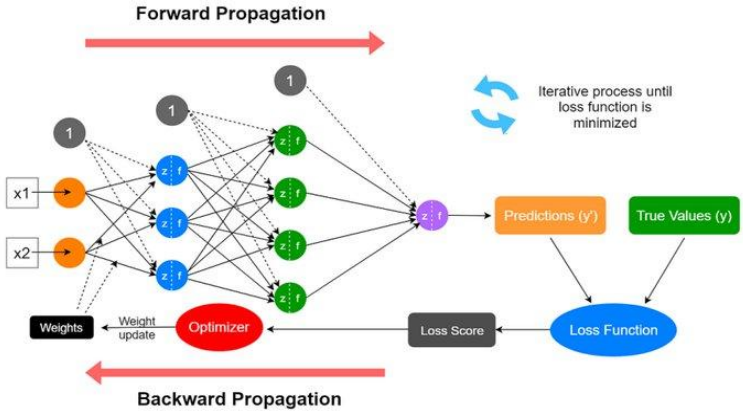

Figure 2:Mécanisme de rétropropagation du gradient.

Pour l'imagerie médicale, le choix du squelette (backbone) du CNN influence directement la qualité des caractéristiques extraites. Le tableau ci-dessous compare les architectures les plus utilisées :

Architecture	Paramètres	Complexité	Efficacité CBIR	Efficacité Siamois
VGG16	~138 M	Très Élevée	Faible (Vecteurs denses)	Moyenne (Sur-apprentissage)
ResNet50	~25.5 M	Moyenne	Très Bonne (Skip Conn.)	Excellente (Stabilité)
DenseNet121	~8.0 M	Faible	Excellente (Gradients)	Très Bonne (Propag. Fine)
EfficientNet	~5.3 M (B0)	Très Faible	Maximale (Multi-échelle)	Excellente (Convergence)

Tableau 1: *Comparatif des architectures de réseaux CNN [Ref ?]*

1.2.2.1 *DenseNet121*

DenseNet121 est une architecture de réseau de neurones convolutif appartenant à la famille des DenseNet. Elle se caractérise par des connexions denses entre les couches, où chaque couche reçoit les informations produites par les couches précédentes. Cette organisation favorise la réutilisation des caractéristiques, améliore la propagation du gradient et permet d'entraîner des réseaux profonds de manière plus efficace.

1.2.2.2 *TorchXRayVision DenseNet121*

TorchXRayVision DenseNet121 est une variante de DenseNet121 intégrée dans la bibliothèque TorchXRayVision, spécialisée dans l'analyse des images médicales. Ce modèle est généralement pré-entraîné sur de grandes bases de radiographies, ce qui lui permet d'extraire des caractéristiques visuelles adaptées aux images X-ray. Il est souvent utilisé comme modèle de base pour des tâches telles que la classification, l'extraction de caractéristiques ou la comparaison d'images médicales.

1.2.2.3 *EfficientNet-B0*

EfficientNet-B0 est une architecture de réseau de neurones convolutif développée pour obtenir un bon équilibre entre la performance et le coût de calcul. Elle repose sur le principe du compound scaling, qui ajuste de manière équilibrée la profondeur, la largeur et la résolution du réseau. EfficientNet-B0 représente la version de base de la famille EfficientNet

1.2.2.4 *ResNet-50*

ResNet-50 est une architecture CNN développée pour faciliter l'entraînement des réseaux profonds. Elle repose sur le principe des connexions résiduelles, qui permettent de transmettre l'information entre les couches et de réduire les problèmes liés à la profondeur du réseau. ResNet-50 est une version composée de 50 couches, largement utilisée dans les tâches de classification et d'extraction de caractéristiques.

1.3 Apprentissage par transfert (Transfer Learning)

L'apprentissage par transfert (Transfer Learning) consiste à réutiliser un modèle déjà entraîné sur une grande base de données afin de l'adapter à une nouvelle tâche. Au lieu d'entraîner un réseau depuis zéro, on exploite les connaissances déjà apprises par le modèle, comme les contours, les formes ou les textures.

Cette approche est utile lorsque les données disponibles sont limitées, car elle permet de réduire le temps d'entraînement, le coût de calcul et le risque de surapprentissage. Dans les CNN, les premières couches apprennent généralement des caractéristiques simples et générales, tandis que les couches profondes apprennent des représentations plus spécifiques.

Deux stratégies sont souvent utilisées :

a) *Extraction de caractéristiques (Feature Extraction) :* On gèle (freeze) l'intégralité de la partie convolutive pré-entraîné (l'extracteur de caractéristiques) ses poids ne sont pas modifiés durant l'entraînement. Seule la tête de décision (classifieur) ajoutée au sommet est entraînée sur le nouveau jeu (dans notre cas dataset des images médicales). Cette approche est indiquée quand les données médicales disponibles sont très peu nombreuses : elle limite drastiquement le nombre de paramètres à optimiser et réduit le risque de sur-apprentissage.

b) Fine-tuning (ajustement fin) : Le fine-tuning, ou ajustement fin, est une technique d'apprentissage automatique qui consiste à réutiliser un modèle déjà pré-entraîné sur une grande quantité de données, puis à l'adapter à une nouvelle tâche plus spécifique(souvent avec un dataset à petite taille). Au lieu d'entraîner un modèle depuis le début, on conserve une partie de connaissances déjà apprises (la moins profond, des connaissances génériques) et on ajuste certaines couches du réseau avec un nouveau jeu de données. Cette méthode permet de gagner du temps, de réduire le besoin en données et d'améliorer les performances sur une tâche ciblée.

1.4 Approche CBIR en imagerie médicale

La Recherche d'Images par le Contenu (Content-Based Image Retrieval, CBIR) est une technique qui permet d'indexer et de retrouver des images en se basant sur leurs caractéristiques visuelles calculées automatiquement [4], sans dépendre d'annotations textuelles subjectives. Appliquée à la radiologie, elle offre une alternative aux systèmes de classification en fournissant au praticien un ensemble de cas cliniques antérieurs visuellement similaires [4], [6], au cas qu'il examine, plutôt qu'un simple label diagnostique.

Un système CBIR s'articule autour de deux phases complémentaires : une phase hors-ligne d'indexation et une phase en ligne de recherche des images similaires [4].

1. **Phase hors-ligne (off-line) :** chaque image de la base historique est passée dans le réseau d'extraction pour produire un vecteur de caractéristiques. Ces vecteurs sont ensuite indexés dans une base vectorielle, par exemple avec FAISS, afin d'accélérer la recherche.
2. **Phase en ligne (on-line) :** à la réception d'une image requête, le système extrait son vecteur de caractéristiques, le compare avec les vecteurs déjà indexés de la base de recherche, puis retourne les Top-K cas les plus proches à l'image requête.

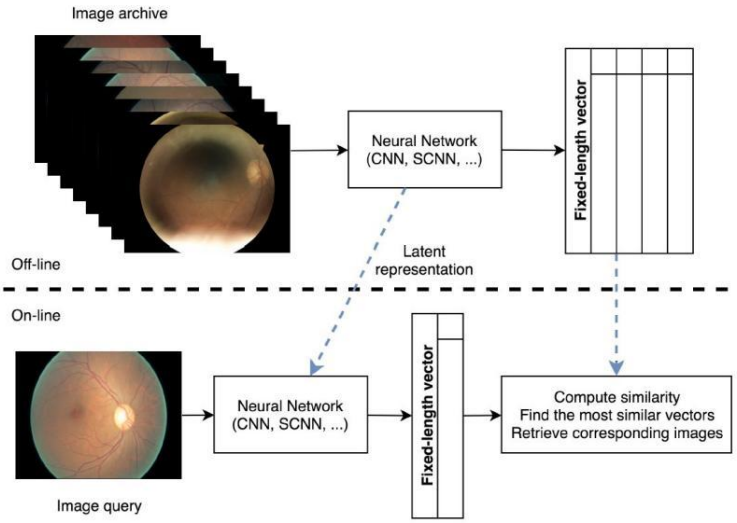

Figure 3:*Architecture fonctionnelle globale d'un système CBIR.*

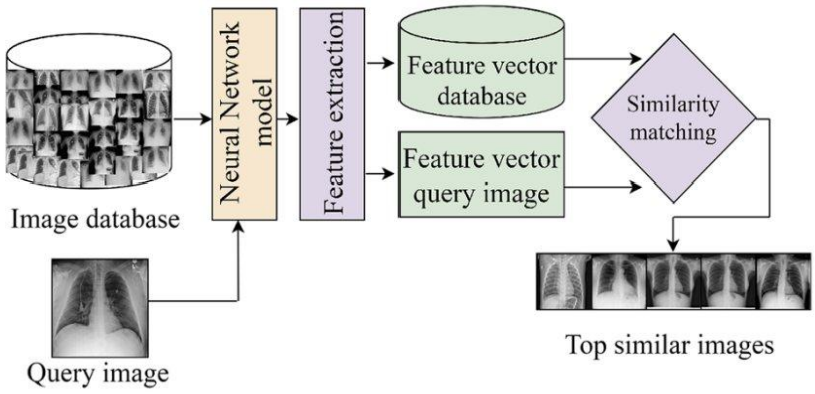

Figure 4:*Application du CBIR aux radiographies thoraciques.*

1.5 Réseaux siamois

Un réseau siamois est composé de deux sous-réseaux identiques qui partagent exactement les mêmes poids [1]. Cette symétrie garantit que deux images d'entrée distinctes sont projetées de la même façon dans l'espace latent, ce qui rend leur comparaison équitable et dépourvue de biais architectural. Plutôt que de produire une étiquette de classe, le réseau prend en entrée une paire ou un triplet d'images et calcule la distance entre leurs représentations vectorielles pour déterminer leur degré de similarité [1], [3].

Chaque sous-réseau joue le rôle d'un extracteur de caractéristiques qui transforme l'image en un vecteur de caractéristiques [4], [5]. Il transforme l'image brute en un vecteur de caractéristiques compact qui capture ses propriétés visuelles essentielles. Ces deux vecteurs sont ensuite comparés à l'aide d'une mesure de distance comme la distance euclidienne ou la similarité cosinus afin de produire un score de similarité.

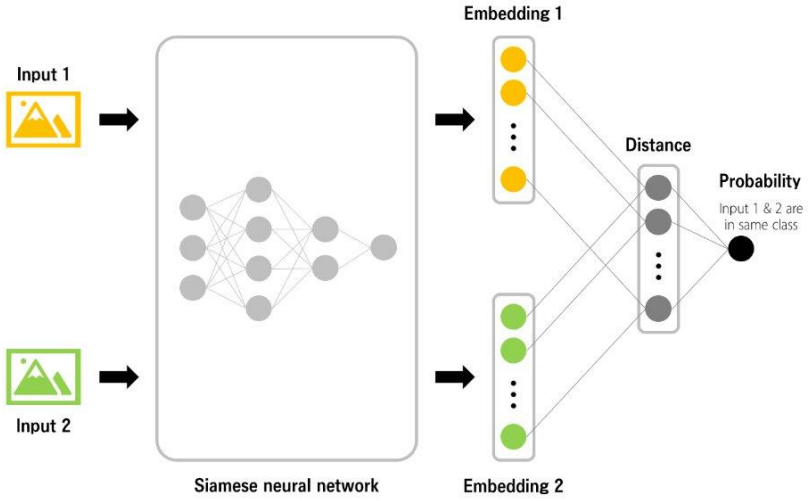

Figure 5:*Architecture fonctionnelle d'un réseau de neurones siamois.*

1.5.1 Fonctions de perte

L'entraînement d'un réseau siamois repose sur des fonctions de perte spécifiques qui permettent d'organiser l'espace de caractéristiques. Leur objectif est de rapprocher les images similaires et d'éloigner les images différentes. La Contrastive Loss est utilisée avec des paires d'images [2]. Elle compare deux descripteurs (vecteurs de caractéristiques) et pénalise le modèle lorsque deux images similaires sont éloignées ou lorsque deux images différentes sont trop proches. Cette fonction utilise une marge afin d'imposer une distance minimale entre les paires différentes.

$$L = (1 - Y).\frac{1}{2}.D^2 + Y.\frac{1}{2}max(0, m - D)^2 \qquad (1)$$

La Triplet Loss utilise trois images : une image ancre A, une image positive P similaire à l'ancre, et une image négative N différente de l'ancre [3]. Son objectif est de rendre l'ancre plus proche de l'image positive que de l'image négative, avec une marge de sécurité α.

$$L(A, P, N) = max(0, d(A, P) - d(A, N) + m) \qquad (2)$$

Hard mining : Le hard mining est une stratégie utilisée pour sélectionner les exemples les plus difficiles pendant l'entraînement. Avec la Contrastive Loss, il permet de choisir les paires positives et négatives les plus difficiles. Avec la Triplet Loss, il permet de sélectionner des triplets plus informatifs, notamment des images négatives proches de l'image ancre dans l'espace de caractéristiques. Cette stratégie rend l'apprentissage plus exigeant et améliore la séparation entre les images similaires et dissimilaires.

1.6 L'apprentissage métrique

L'apprentissage métrique consiste à apprendre une représentation dans laquelle les données similaires sont proches et les données différentes sont éloignées [2], [3]. Dans le cas des réseaux siamois, chaque image est transformée en un vecteur de caractéristiques [4]. La comparaison entre deux images se fait ensuite en mesurant la distance ou la similarité entre leurs vecteurs de caractéristiques.

Deux mesures sont couramment utilisées :

Distance euclidienne : mesure l'écart direct entre deux vecteurs dans l'espace d'embedding. Plus la distance est faible, plus les deux images sont considérées comme similaire.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \qquad (3)$$

Similarité cosinus : mesure l'angle entre deux vecteurs. Plus la valeur est proche de 1, plus les deux embeddings sont orientés dans la même direction, donc plus les images sont similaires.

$$\textit{Similiratié Cosinus} = \frac{A.B}{\|A\|.\|B\|} = \frac{\sum_{i=1}^{n} AiBi}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \qquad (4)$$

1.7 Conclusion

Ce premier chapitre a établi les fondements théoriques de notre système. En partant de la hiérarchie de l'IA jusqu'aux métriques d'évaluation, en passant par les CNN, le Transfer Learning, le CBIR et les réseaux siamois, nous avons montré en quoi l'apprentissage par similarité représente une réponse adaptée aux contraintes de l'imagerie médicale. Le chapitre suivant présentera la conception technique du système et les choix d'implémentation retenus.

2 Chapitre 2 : Approches élaborées pour la recherche d'images radiologiques similaires

2.1 Introduction

Dans ce chapitre, nous expliquons les différentes approches et techniques appliquées à plusieurs architectures CNN, afin de réaliser un système permettant de rechercher des radiographies pulmonaires similaires à partir d'une image requête.

2.2 Méthodologie

Dans ce travail, nous explorons différents modèles d'apprentissage profond pour la recherche de radiographies pulmonaires similaires à l'aide des architectures CNN et de réseaux siamois. Nous étudions d'abord la capacité de transfert des caractéristiques apprises sur ImageNet vers les images radiologiques, puis nous exploitons également TorchXRayVision, une bibliothèque spécialisée dans les modèles et datasets de radiographies thoraciques.

Pour atteindre notre objectif, nous avons utilisé DenseNet121 pré-entraîné sur ImageNet comme modèle de base, puis TorchXRayVision DenseNet121, EfficientNet-B0 et ResNet-50 avec différentes stratégies d'entraînement. Les modèles sont entraînés avec Contrastive Loss ou Triplet Loss, parfois associée au Visual Hard Mining et au fine-tuning progressif, afin d'apprendre une caractérisation pertinente, permettant de rapprocher les images similaires et d'éloigner les images dissimilaires dans l'espace de caractéristiques.

Les évaluations utilisées sont : Precision@K, Recall@K, mAP@K, NDCG@K , SSIM et l'analyse visuelle des résultats Top-5. La meilleure configuration est ensuite utilisée dans l'application finale pour rechercher les cas les plus similaires à une image requête.

2.3 Prétraitement des données

Avant l'entraînement, nous avons utilisé OpenCV et PIL pour préparer les radiographies pulmonaires. Les images sont d'abord lues en niveaux de gris, puis une amélioration du contraste est appliquée à l'aide de la méthode CLAHE. Ensuite, un léger filtrage est utilisé afin de réduire le bruit et d'obtenir des images plus homogènes.

Lorsque les masques pulmonaires sont disponibles, ils sont utilisés pour isoler la région pulmonaire et réduire l'influence des zones non pertinentes de l'image. Les radiographies sont ensuite redimensionnées à 224 × 224 pixels afin d'être compatibles avec le backbone utilisé. Dans le cas de TorchXRayVision DenseNet121, les images en niveaux de gris sont normalisées selon le format attendu par ce modèle, avec une mise à l'échelle adaptée aux radiographies thoraciques.

Ce prétraitement permet d'uniformiser les images avant leur passage dans le réseau siamois et d'améliorer la qualité des caractéristiques extraites.

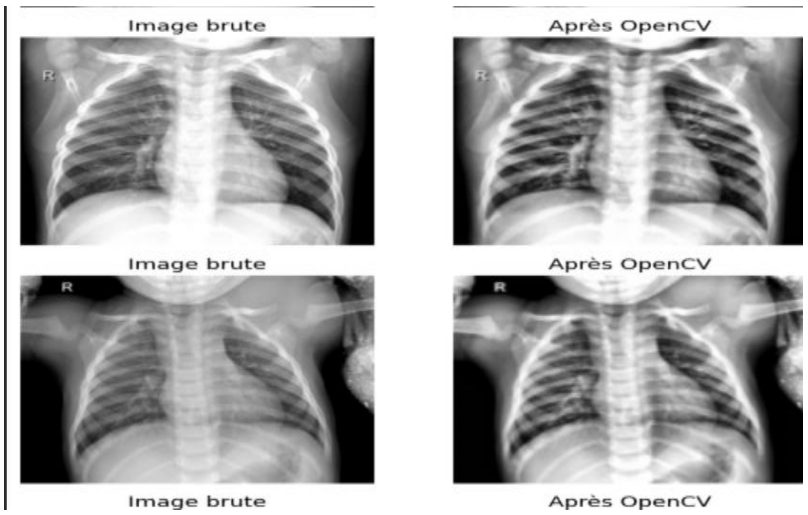

Figure 6:*Exemples de radiographies pulmonaires avant et après prétraitement (CLAH- OpenCV).*

2.4 Indexation des vecteurs de caractéristique avec FAISS

Nous avons ensuite utilisé FAISS pour créer un index vectoriel à partir des vecteurs de caractéristiques extraites. Cet index permet d'accélérer la recherche des radiographies les plus similaires à une image requête. Lorsqu'une nouvelle radiographie est fournie, son vecteur de caractéristiques est calculé, puis comparé aux vecteurs de caractéristiques stockés dans l'index à l'aide de la similarité cosinus. Le système retourne enfin les Top-K cas les plus proches, sans filtrage par vraie classe, afin de simuler une recherche CBIR globale.

2.5 Approches proposées

Dans cette partie, nous présentons les Trois approches réalisées. Chaque expérimentation a été conçue pour répondre à un objectif précis. La première sert de base de comparaison, la deuxième étudie l'effet des fonctions de pertes et hard mining , la troisième introduit le fine-tuning progressif avec Triplet Loss avec autre backbones plus léger

2.5.1 Approche 1 : Niveaux de fine-tuning sur le modèle DenseNet121 pré-entraîné sur ImageNet

Dans cette première approche, nous utilisons le modèle DenseNet121 avec des poids pré-entraînés sur ImageNet. L'apprentissage est réalisé avec la Contrastive Loss. Deux scénarios sont alors étudiés :

2.5.1.1 Scénario 1 : Sans fine-tuning (mode extracteur de caractérstique)

Dans ce scénario appelé «baseline» l'extracteur de DenseNet121 reste gelé et seule la tête d'embedding est entraînée. Cela permet d'évaluer les caractéristiques générales apprises sur ImageNet sans adaptation du backbone.

2.5.1.2 Scénario 2 : mode fine-tuning

Dans ce scénario, les couches finales du modèle, principalement denseblock4 et norm5, sont dégelées puis réentraînées avec la tête d'embedding. Cela permet d'adapter le modèle aux radiographies pulmonaires.

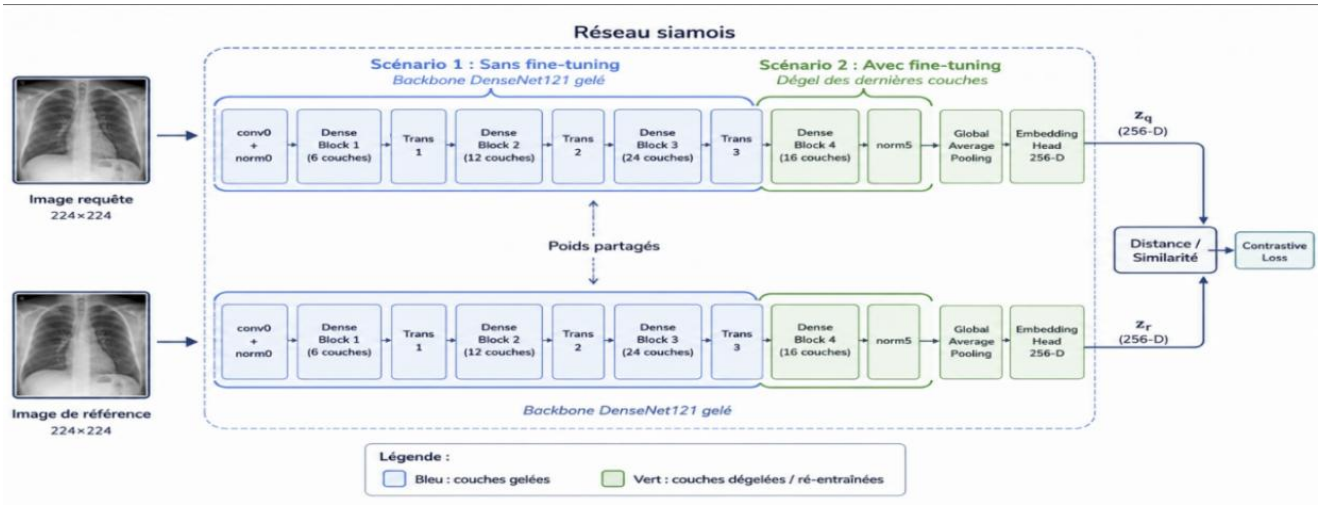

Figure 7: *Architecture de l'expérimentation 1.*

2.5.2 Approche 2 : Comparaison des fonctions de perte avec TorchXRayVision DenseNet121 et Visual Hard Mining

Dans cette expérimentation, nous utilisons TorchXRayVision DenseNet121, un modèle pré-entraîné sur des radiographies thoraciques. L'objectif est de comparer l'impact de deux fonctions de perte sur la qualité des vecteurs de caractéristiques générés pour la recherche de radiographies similaires. Dans les deux scénarios, le Visual Hard Mining est utilisé afin de sélectionner les exemples les plus difficiles pendant l'entraînement. La partie extracteur de caractéristiques reste gelé seule la tête d'embedding est entraînée afin de produire des vecteurs adaptés à la recherche d'images similaires.

2.5.2.1 Scénario 1 : Contrastive Loss

Dans ce premier scénario Le modèle reçoit des paires d'images : des paires positives composées d'images similaires et des paires négatives composées d'images différentes.

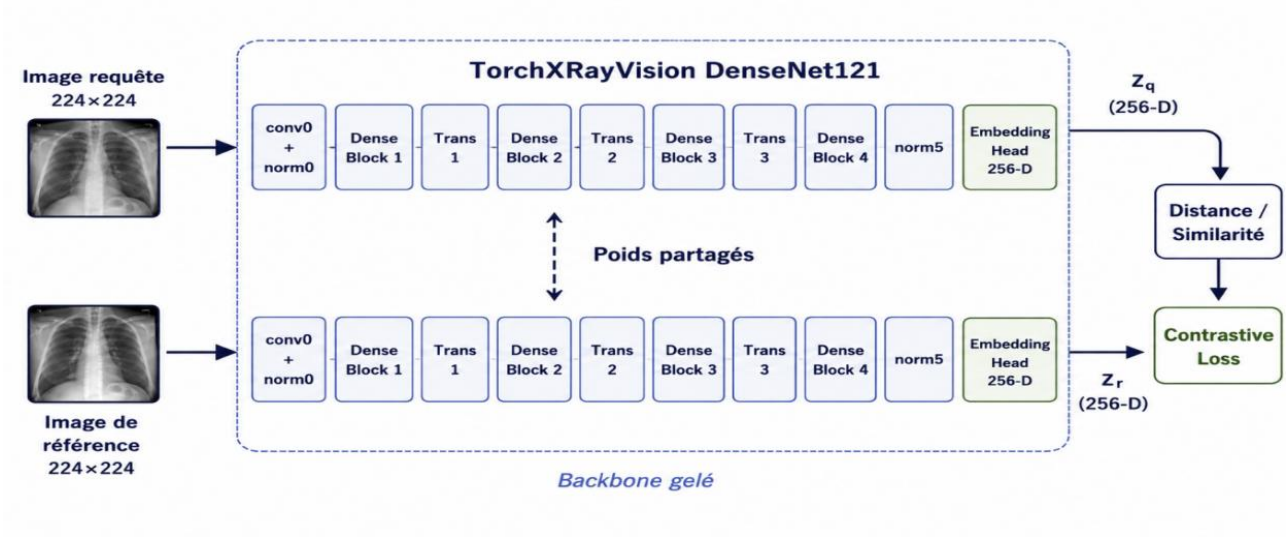

Figure 8: *Architecture de l'expérimentation 2 Scénario1.*

2.5.2.2 Scénario 2 : Triplet loss

Dans ce scénario, L'entraînement est réalisé à partir de triplets d'images : une image ancre, une image positive similaire à l'ancre et une image négative différente qui permet ainsi d'apprendre à extraire des caractéristiques pertinentes, où les images similaires sont rapprochées, tandis que les images différentes sont éloignées..

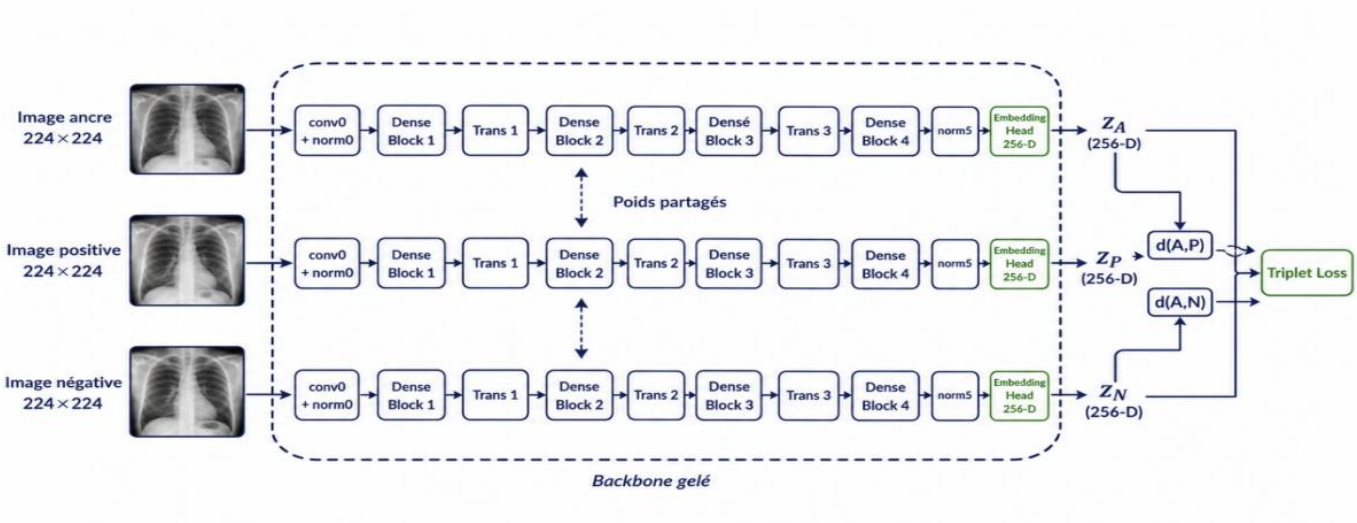

Figure 9: *Architecture de l'expérimentation 2 Scénario2.*

2.5.3 Approche 3 : Configuration avec différents backbones selon la meilleure configuration des aproches précédentes

Dans cette expérimentation, nous reprenons la meilleure configuration obtenue dans les expérimentations précédentes, basée sur la Triplet Loss, le hard mining et le fine-tuning progressif. Cette configuration est ensuite appliquée à trois backbones différents : TorchXRayVision DenseNet121, EfficientNet-B0 et ResNet-50.

2.5.3.1 Scénario 1 : TorchXRayVision DenseNet121

Dans ce scénario, le modèle commence avec le backbone TorchXRayVision DenseNet121 gelé, puis dans une deuxième phase, les couches denseblock4 et norm5, sont dégelées. Ces couches sont ensuite réentraînées avec la tête d'embedding en utilisant un faible taux d'apprentissage.

2.5.3.2 Scénario 2 : EfficientNet-B0

Dans ce scénario, le modèle commence avec un extracteur gelé, puis dans une deuxième phase, les blocs supérieurs de l'extracteur d'EfficientNet-B0 sont dégelés, principalement les blocs : 6 , 7 et 8. Ces blocs sont ensuite réentraînés avec la tête d'embedding en utilisant un faible taux d'apprentissage.

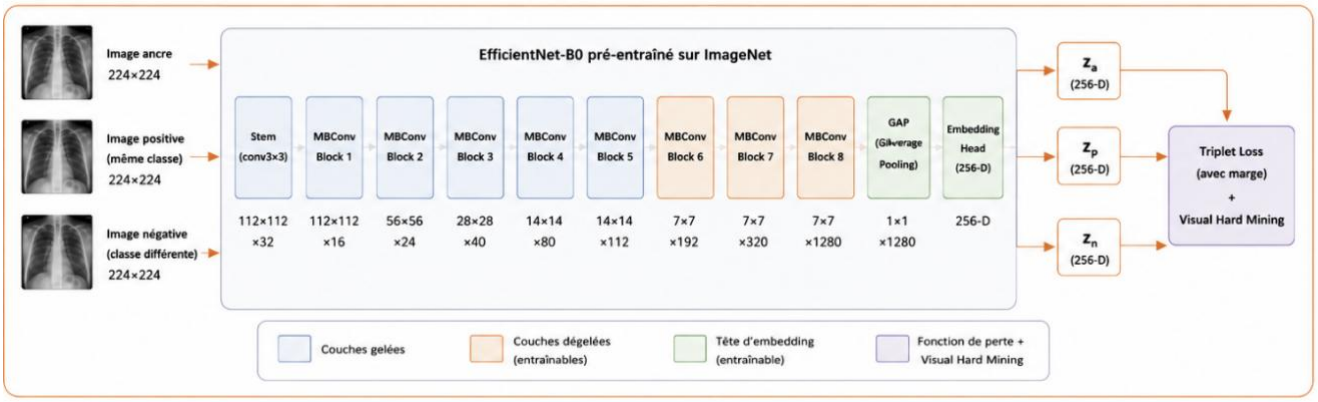

Figure 10: *Architecture de l'expérimentation 3 Scénario2.*

2.5.3.3 Scénario 3 : ResNet-50

Dans ce scénario, le backbone ResNet-50 est d'abord gelé, et seule la tête d'embedding est entraînée. Dans une deuxième phase, le dernier bloc du réseau, principalement layer4, est dégelé. Ce bloc est ensuite réentraîné avec la tête d'embedding en utilisant un faible taux d'apprentissage.

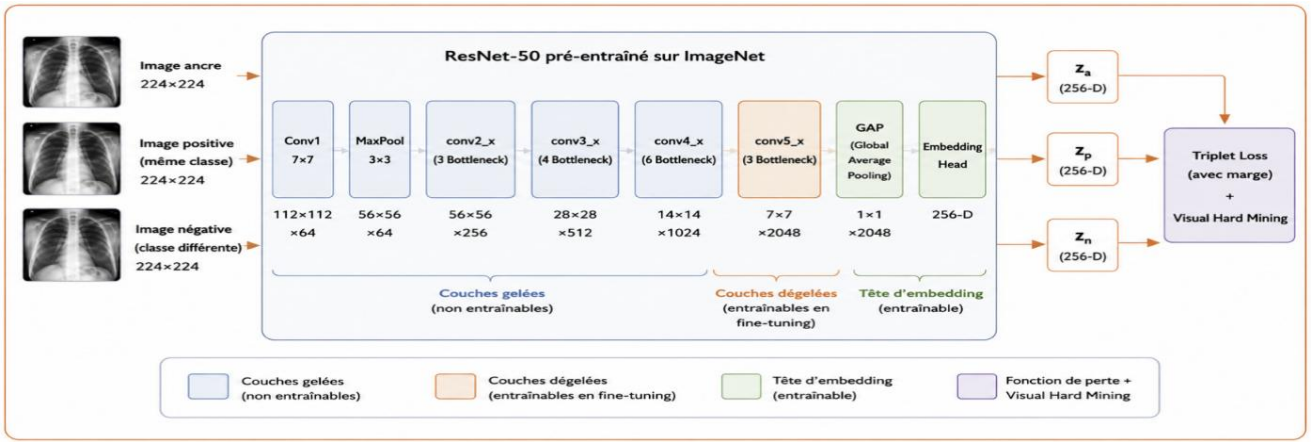

Figure 11:*Architecture de l'expérimentation 3 Scénario3.*

2.6 Conclusion

Dans ce chapitre, nous avons présenté les approches élaborées pour la recherche de radiographies pulmonaires similaires. Nous avons décrit la méthodologie générale, le prétraitement des images, l'extraction des de caractéristiques ainsi que l'indexation avec FAISS. Nous avons ensuite présenté les principales expérimentations : DenseNet121 pré-entraîné sur ImageNet, TorchXRayVision DenseNet121 avec différentes fonctions de perte, puis EfficientNet-B0 et ResNet-50 afin de comparer l'effet de différents extracteurs de caractéristiques.

Ces approches permettent d'évaluer l'impact du fine-tuning, du choix du backbone et de la fonction de perte sur la qualité de la recherche d'images similaires. Le chapitre suivant présentera les résultats obtenus, l'évaluation des modèles ainsi que l'interface de l'application developpée.

3 Chapitre 3 : Expérimentations, résultats et discussion

3.1 Introduction

Dans ce chapitre, nous présentons l'implémentation de notre système d'assistance au pré-diagnostic radiologique par recherche d'images similaires. Nous décrivons l'environnement de développement, les outils utilisés, le dataset utilisé dans nos différentes expérimentations des trois approches élaborées, ainsi que les métriques choisies pour évaluer les performances des différents modèles.

Nous présentons ensuite les résultats expérimentaux obtenus à travers les approches proposées, notamment l'utilisation de DenseNet121, TorchXRayVision DenseNet121, EfficientNet-B0 et ResNet-50 avec différentes stratégies d'apprentissage de similarité. Enfin, nous terminons par une présentation de l'interface graphique de notre application web.

3.2 Environnement de développement

3.2.1 Environnement matériel

Afin de mener nos expérimentations, nous avons utilisé un ordinateur personnel pour le développement local, ainsi qu'un environnement Cloud, notamment Kaggle Notebook pour l'entraînement des modèles. Les caractéristiques de la machine utilisée pour le développement local sont les suivantes :

- Processeur : Intel(R) Core(TM) Ultra 7 155H, 3.80 GHz
- Mémoire RAM installée : 32 Go
- Carte graphique : Intel(R) Arc(TM) Graphics, 128 Mo
- Stockage : 954 Go
- Type du système : système d'exploitation 64 bits, processeur x64
- Système d'exploitation : Windows 11 Professionnel, version 25H2
- GPU utilisé pour l'entraînement : Kaggle Notebook

3.2.2 Outils et langage de développement

Python :Python a été le langage principal du projet. Il a servi à préparer les données, traiter les images, implémenter les réseaux siamois, entraîner les modèles et calculer les métriques d'évaluation.



Figure 12: *Logo de Python*

PyTorch : PyTorch est une bibliothèque open source destinée au Deep Learning. Elle permet de construire, entraîner et tester des réseaux de neurones de manière flexible, avec un support important pour le calcul sur GPU.



Figure 13*:Logo de PyTorch.*

Kaggle Notebook : Kaggle a été utilisé comme environnement de travail en ligne pour accéder facilement au dataset COVID-19 Radiography Database et exécuter les entraînements avec des ressources GPU.



Figure 14*:Logo de Kaggle*

Flask : Flask est un micro-framework web écrit en Python. Il permet de créer des applications web de manière simple et flexible, avec la possibilité de gérer les routes, les pages HTML et les requêtes utilisateur.



***Figure 15**:Logo de Flask*

OpenCV : OpenCV est une bibliothèque open source spécialisée dans la vision par ordinateur et le traitement d'images. Elle propose plusieurs fonctions pour lire, modifier, analyser et transformer des images.



***Figure 16**:Logo OpenCv*

PyCharm : PyCharm est un environnement de développement intégré, appelé IDE, conçu pour programme. Python. Il offre plusieurs fonctionnalités comme l'édition de code, le débogage, l'analyse du code et l'intégration avec des outils de gestion de projet.



***Figure 17**:Logo Pycharm*

3.3 Description de la base de données utilisée

Le COVID-19 Radiography Database est un dataset de radiographies pulmonaires publié sur Kaggle. Il contient environ 21 165 images réparties en quatre classes : COVID-19, Normal, Lung Opacity et Viral Pneumonia. Cette base est utilisée pour entraîner et évaluer les modèles de recherche d'images similaires. Les images sont exploitées pour former des paires et des triplets nécessaires à l'apprentissage des réseaux siamois. La base est divisée en ensembles d'entraînement, de validation et de test afin d'évaluer les performances des modèles de manière correcte.

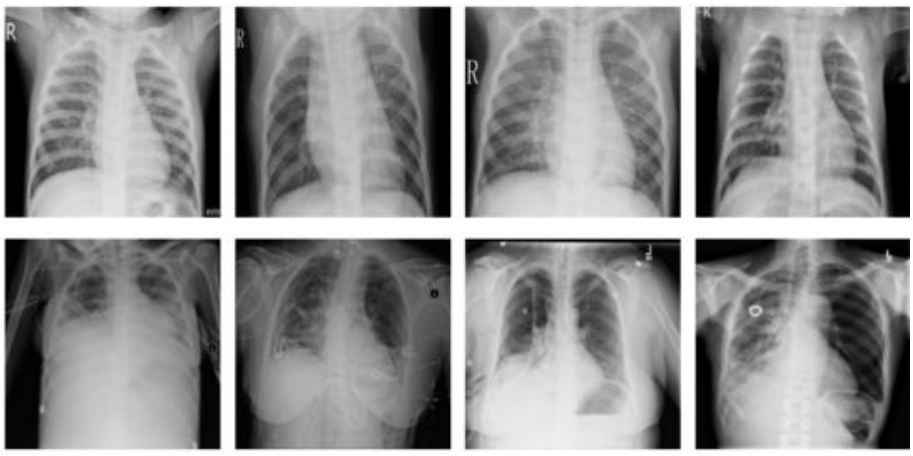

***Figure 18**: Echantilons d'images de dataset : COVID-19 Radiography Database*

3.4 Métriques d'évaluation

Pour évaluer notre système, nous avons utilisé des métriques adaptées à la recherche d'images similaires. L'objectif est de mesurer la qualité des images retournées dans les premiers résultats. Les métriques utilisées sont :

- Precision@K : proportion d'images pertinentes parmi les K premiers résultats retournés.
- Recall@K : proportion d'images pertinentes retrouvées par le système parmi l'ensemble des images pertinentes disponibles.
- Hit Rate@K : indique si au moins une image pertinente est présente dans les K premiers résultats.
- mAP@K : mesure la qualité globale du classement des résultats pertinents dans les K premières positions.
- MRR@K : mesure la position du premier résultat pertinent ; plus sa valeur est élevée, plus le premier résultat pertinent apparaît tôt.
- nDCG@K : évalue la qualité du classement en donnant plus d'importance aux images pertinentes placées dans les premiers rangs.

3.5 Expérimentations et résultats :

L'apprentissage des modèles dans nos expérimentations est réalisé sur 30 époques, un choix adapté à la taille du dataset et au coût d'entraînement des réseaux siamois. Afin d'éviter le surapprentissage et de conserver le meilleur modèle obtenu, nous avons utilisé les mécanismes suivants :

Arrêt précoce (Early Stopping) : l'entraînement est arrêté si la performance sur l'ensemble de validation ne s'améliore pas pendant plusieurs époques successives avec sauuvegardation du meilleur modèle.

Réduction du taux d'apprentissage : le taux d'apprentissage est réduit lorsque la performance de validation stagne, afin de permettre un ajustement plus fin des poids du modèle.

3.5.1 Expérimentation 1 (Approche 1)

Dans cette expérimentation, nous avons utilisé le modèle DenseNet121 pré-entraîné sur ImageNet.

3.5.1.1 Scénario 1 : le modèle DenseNet121 Sans fine-tuning

Dans ce scénario l'apprentissage montre une diminution progressive de la perte d'entraînement et de validation, ce qui indique une évolution stable du modèle. La séparation entre les paires positives et négatives s'améliore également. Le mAP@5 augmente rapidement au début de l'entraînement puis se stabilise autour de 0,74 , ce qui représente une première performance acceptable sans adaptation du backbone.

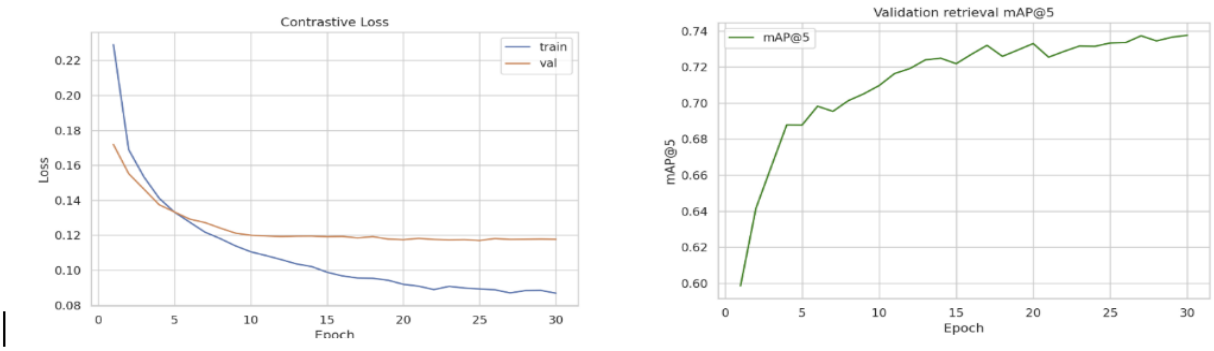

Figure 19: *Courbes d'apprentissage du scénario 2 de l'expérimentation 1.*

3.5.1.2 Scénario 2 : le modèle DenseNet121 Avec fine-tuning

Dans ce scénario, le backbone DenseNet121 est d'abord gelé, puis les dernières couches sont dégelées à partir de l'époque 6. Les courbes montrent une diminution de la perte et une bonne séparation entre les paires similaires et différentes. Le mAP@5 augmente plus que scenario 1 autour de 0,77, ce qui représente une meilleure performance avec le fine tuning

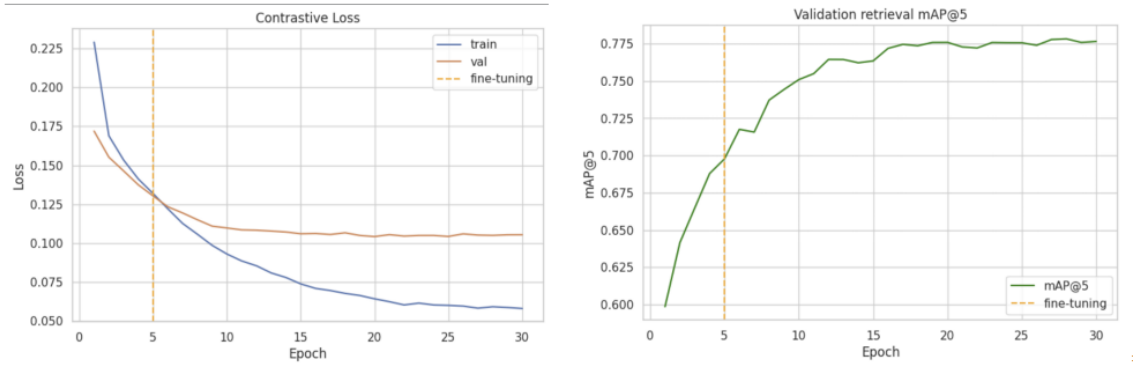

Figure 20 : *Courbes d'apprentissage du scénario 2 de l'expérimentation 1.*

Metrique dans test set/ scenario	**Precision@5**	**mAP@5**	**nDCG@5**	**Recall@5**	**Hit Rate@5**	**MRR@5**
Scenario 1	82,63%	78,82%	82,69%	0,091	94,47%	87,65%
Scenario 2	**84,87%**	**81,66%**	**85,00%**	**0,094%**	**95,04%**	**89,41%**

Tableau 1: *Comparatif des métriques d'évaluation sur l'ensemble de test pour l'expérimentation 1*

Le scénario avec fine-tuning obtient de meilleurs résultats que le scénario sans fine-tuning. La Precision@5 passe de 82,63 % à 84,87 %, tandis que le mAP@5 augmente de 78,82 % à 81,66 %.Ces résultats montrent que le fine-tuning des dernières couches du backbone améliore la qualité des embeddings et permet de retrouver des radiographies plus pertinentes parmi les premiers résultats.

3.5.2 Expérimentation 2 (Approche 2)

Dans les expériences qui suivent, nous avons utilisé le modèle TorchXRayVision DenseNet121.

3.5.2.1 Scénario 1 : TorchXRayVision DenseNet121 avec Contrastive Loss et Visual Hard Mining

Les courbes montrent une évolution stable de l'apprentissage : la perte d'entraînement diminue progressivement, tandis que la perte de validation reste relativement stable après les premières époques. La séparation entre les paires positives et négatives s'améliore également, avec une diminution des distances positives et une augmentation des distances négatives. Le modèle atteint un mAP@5 d'environ 0,69 sur l'ensemble de validation, ce qui indique qu'il retourne des résultats pertinents parmi les cinq premières images similaires.

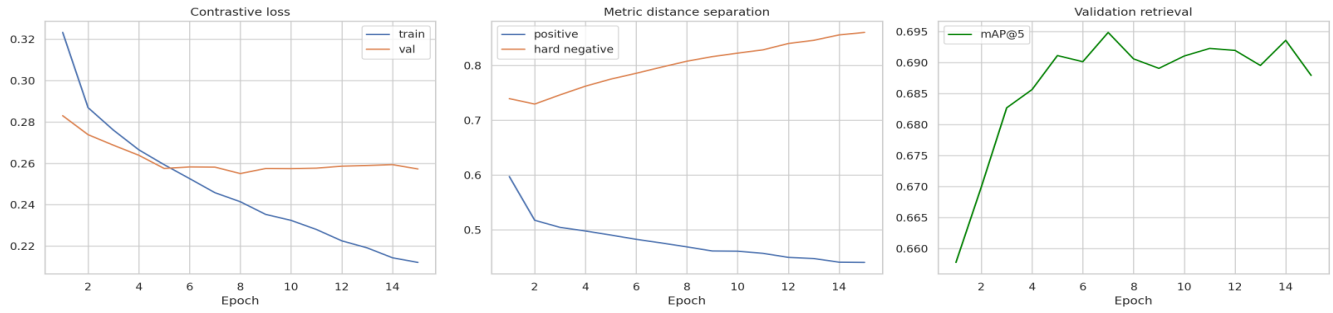

Figure 21 : *Courbes d'apprentissage du scénario 1 de l'expérimentation 2.*

3.5.2.2 Scénario 2 : TorchXRayVision DenseNet121 avec Triplet Loss et Visual Hard Mining

Les courbes montrent une bonne évolution : la perte d'entraînement diminue progressivement et le mAP@5 augmente jusqu'à environ 0,77 sur l'ensemble de validation. Cela montre que le modèle arrive à mieux organiser la description d'image et à retourner des images pertinentes parmi les premiers résultats

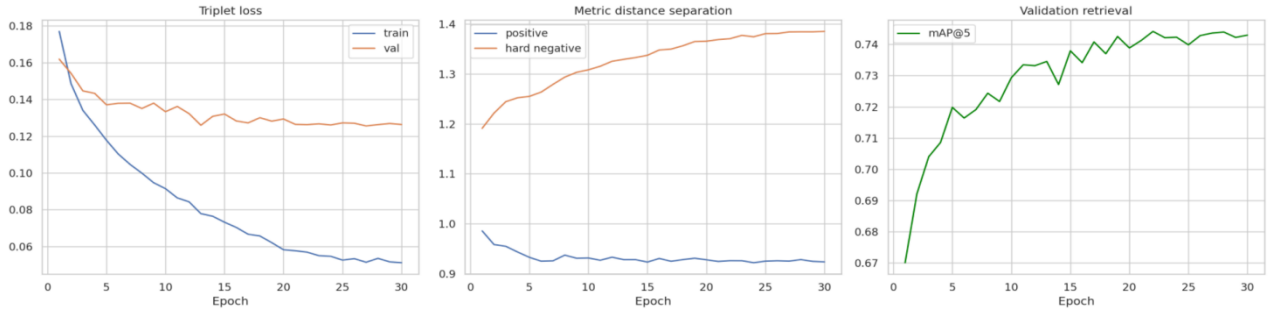

***Figure 22** : Courbes d'apprentissage du scénario 2 de l'expérimentation 2.*

Métriques dans test test / scenario	Precision@5	mAP@5	nDCG@5	Hit Rate@5	Recall@5
Scenario 1	81,53 %	77,50 %	82,11 %	95,84 %	0,084%
Scenario 2	84,82%	81,49%	85,14 %	95,90 %	0,090 %

***Tableau 2:** Comparatif des métriques d'évaluation sur l'ensemble de test pour l'expérimentation 2.*

Le scénario basé sur la Triplet Loss obtient de meilleurs résultats que celui basé sur la Contrastive Loss. La Precision@5 passe de 81,53 % à 84,82 %, tandis que le mAP@5 augmente de 77,50 % à 81,49 %. Le nDCG@5 progresse également de 82,11 % à 85,14 %, ce qui montre que les images pertinentes sont mieux placées dans les premiers résultats

3.5.3 Expérimentation 3 (Approche 3)

3.5.3.1 Scénario1: TorchXRayVision DenseNet121

Le modèle converge correctement : la Triplet Loss diminue, tandis que la séparation entre les images positives et négatives devient plus claire. Le mAP@5 atteint environ 0,75 sur l'ensemble de validation, ce qui montre une bonne capacité à retrouver des radiographies similaires. Après plusieurs époques, les performances se stabilisent.

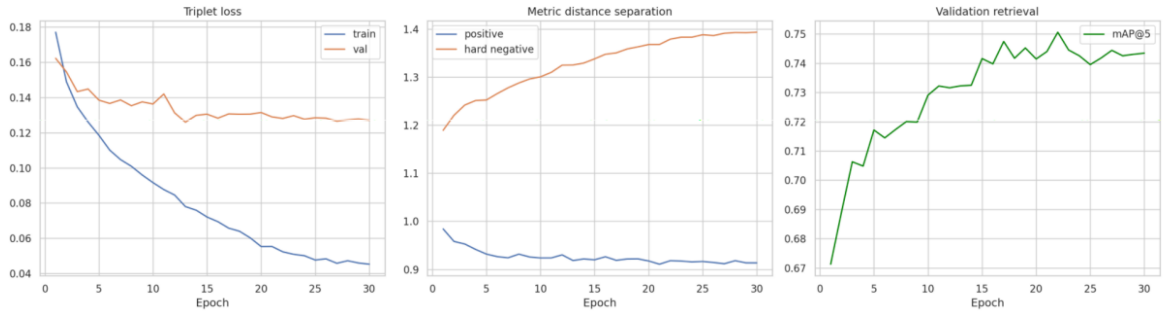

***Figure 23:** Courbes d'apprentissage du scénario 1 de l'expérimentation 3.*

3.5.3.2 Scénario2 : EfficientNet B0

Le modèle montre une évolution satisfaisante durant l'apprentissage. Le score mAP@5 atteint environ 0,77 sur l'ensemble de validation, ce qui confirme l'efficacité de cette approche pour la recherche de radiographies pulmonaires similaires.

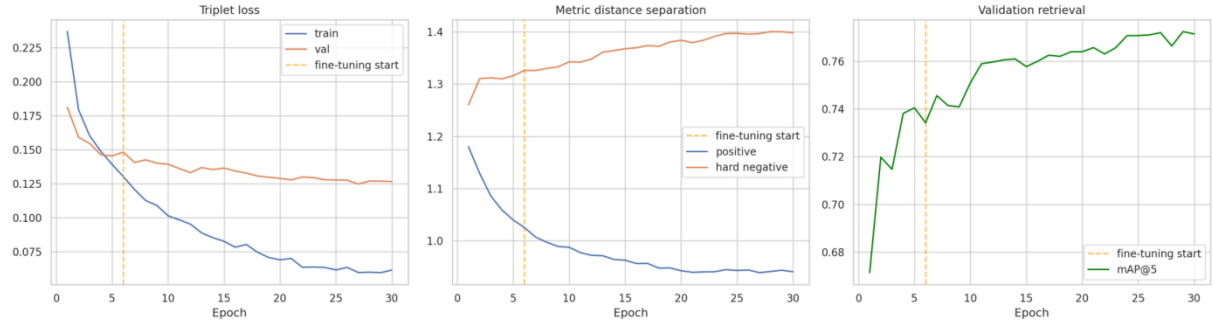

Figure 24:*Courbes d'apprentissage du scénario 2 de l'expérimentation 3.*

3.5.3.3 *Scénario3 : ResNet-50*

Le modèle montre une bonne convergence : la Triplet Loss diminue et la séparation entre images similaires et dissimilaires s'améliore. Le mAP@5 atteint environ 0,77 sur l'ensemble de validation, ce qui indique des résultats satisfaisants pour la recherche d'images similaires.

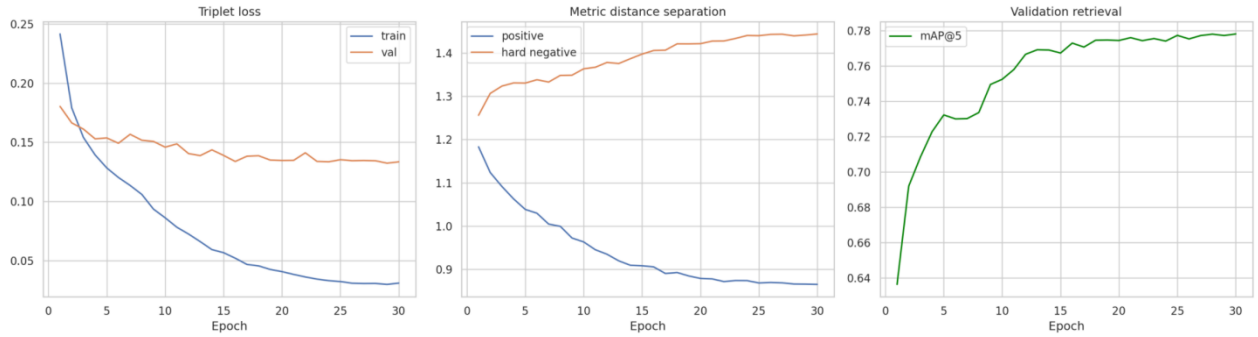

Figure 25: *Courbes d'apprentissage du scénario 3 de l'expérimentation 3.*

Métriques sur test set/scenario	**Precision@5**	**mAP@5**	**nDCG@5**	**Hit Rate@5**	**Recall@5**
Scénario 1	84.72%	81.43%	85.04%	95.61%	0.08995 %
Scénario 2	84.15%	81.22%	84.21%	93.04%	0.093%
Scénario 3	84,88%	82,14%	84,85%	93,43%	0.09%

Tableau 3:*Comparatif des métriques d'évaluation sur l'ensemble de test pour l'expérimentation 3.*

3.6 Discussion générale des résultats

En suivant les différentes expérimentations, nous remarquons que les modèles basés sur l'apprentissage par transfert donnent de meilleurs résultats que le CNN simple entraîné à partir de zéro. L'utilisation de modèles pré-entraînés permet d'extraire des caractéristiques plus riches et plus adaptées aux radiographies pulmonaires.

Nous constatons également que l'utilisation de la Triplet Loss, associée au hard mining, améliore la qualité descriptive. Cette stratégie aide le modèle à mieux rapprocher les images similaires et à éloigner les images dissimilaires, ce qui est essentiel pour la recherche d'images médicales similaires.

Dans l'expérimentation 3, les trois backbones donnent des résultats proches. ResNet-50 obtient le meilleur mAP@5 avec 82,14 %, tandis que TorchXRayVision DenseNet121 obtient le meilleur nDCG@5 avec 85,04 % . Ces résultats montrent que le choix du backbone influence la qualité des embeddings, mais que les trois modèles restent globalement performants pour la recherche de radiographies similaires.

3.7 Analyse visuelle avec Grad-Cam

Afin de mieux interpréter les résultats obtenus, nous avons utilisé Grad-CAM comme méthode d'explication visuelle. Dans notre projet, elle permet de vérifier si le modèle se concentre principalement sur les régions

pulmonaires pertinentes lors de la recherche des cas similaires. Grad-CAM utilise les gradients de la dernière couche convolutionnelle pour produire une carte de localisation visuelle.

3.8 Interface graphique

3.8.1 Vue générale de l'interface graphique de l'application

L'interface graphique joue un rôle important dans la facilité d'utilisation de notre système. Dans ce projet, nous avons développé une application web appelée RadioSim, permettant à l'utilisateur d'importer une radiographie pulmonaire et d'obtenir les cas les plus similaires à partir d'une base d'images

3.8.2 Le développement de l'application

Pour développer l'application, nous avons utilisé Flask et Les interfaces ont été réalisées avec HTML, CSS et JavaScript. La partie Python permet de charger le modèle, prétraiter l'image, extraire l'embedding et rechercher les images similaires.

L'application propose actuellement trois modèles aux choix : TorchXRayVision DenseNet121 , EfficientNet-B0 et Res-Net50.

Page principale : La page principale présente brièvement le système RadioSim et permet à l'utilisateur d'accéder à la recherche.



Figure 26:*page d'accueil de l 'application RadioSim*

Page de sélection du modèle : L'utilisateur peut choisir le modèle de recherche qu'il souhaite utiliser parmi les modèles disponibles.



Figure 27:*page de sélection du modèle*

Page du traitement de l'image et la recherche : Après le choix du modèle, l'utilisateur importe une radiographie pulmonaire. L'image est ensuite prétraitée et envoyée au modèle pour extraire son vecteur de caractéristiques.

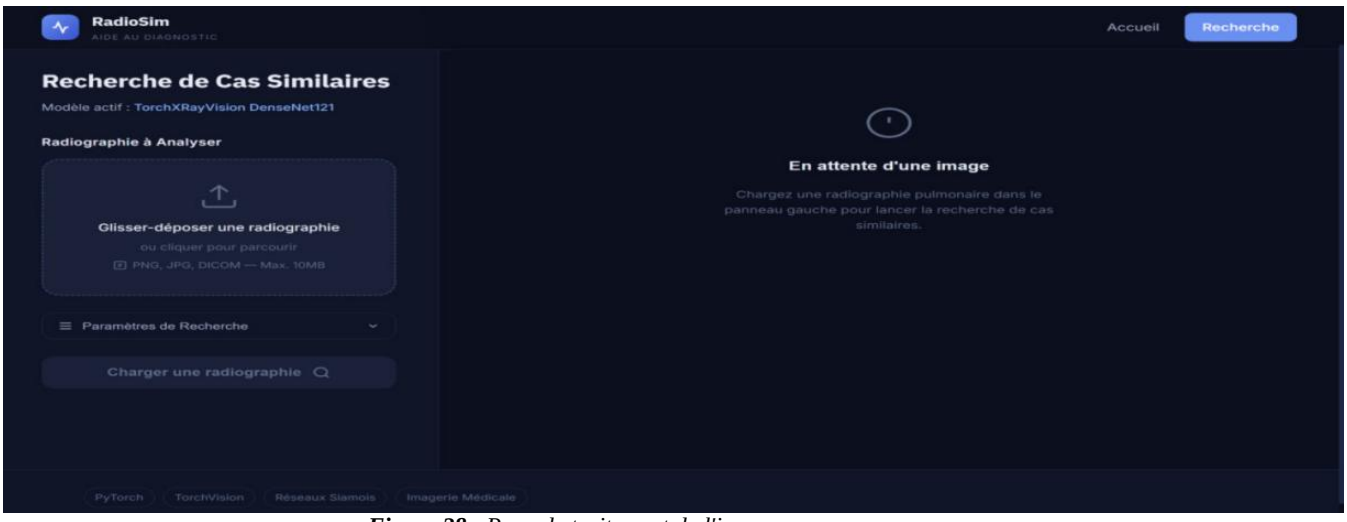

Figure 28 : *Page de traitement de l'image*

Après le traitement, l'application affiche les images les plus similaires trouvées dans la base, avec leurs classes correspondantes et explication Grad-Cam.

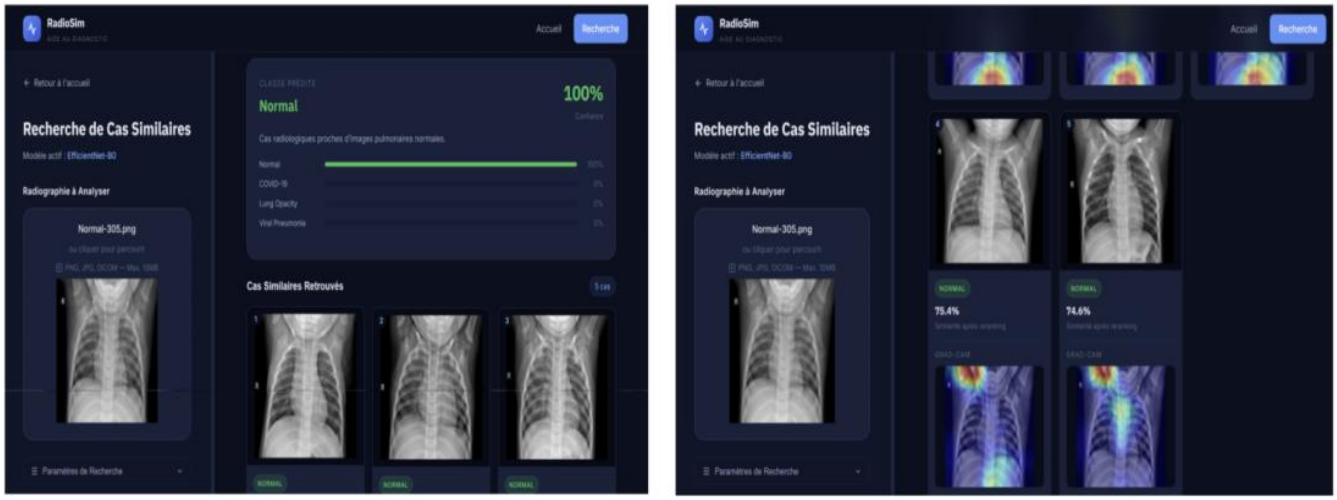

Figure 29 : *Page de résultats avec Grad-cam*

3.9 Conclusion

Nous avons présenté tout au long de ce chapitre l'environnement de travail, les outils utilisés, les résultats des approches testées ainsi que l'interface web de notre application de recherche de radiographies pulmonaires similaires. Les résultats obtenus montrent que l'apprentissage par transfert (fine tuning), la Triplet Loss et les stratégies de hard mining permettent d'obtenir de bonnes performances dans le cadre d'un système CBIR médical.

4 Conclusion générale

La recherche médicale connaît une évolution importante grâce à l'intelligence artificielle et à l'abondance des données médicales. Les techniques modernes permettent d'analyser et de comparer rapidement des images radiologiques, ce qui peut apporter une aide supplémentaire aux spécialistes dans leur travail.

Dans le présent travail, nous avons élaboré trois approches de comparaison d'images basées sur des réseaux siamois et la recherche d'images par le contenu afin de retrouver les radiographies pulmonaires les plus similaires à une image requête. Cette approche repose sur l'extraction de caractéristiques et la comparaison d'images, au lieu d'une simple classification.

Les expérimentations réalisées avec DenseNet121, TorchXRayVision DenseNet121, EfficientNet-B0 et ResNet-50 ont permis d'évaluer l'impact du choix de l'architecture CNN, du mode de fine-tuning et des fonctions de perte sur la qualité des résultats. Les résultats obtenus montrent que le système peut fournir des cas visuellement comparables à l'utilisateur, tout en restant un outil d'aide au pré-diagnostic qui doit être interprété par un spécialiste.

Dans les perspectives de ce travail, nous souhaitons améliorer la qualité des résultats par l'utilisation d'une base plus large, les tranformers de vision, de l'amélioration de la segmentation pulmonaire, l'optimisation de l'indexation FAISS et l'amélioration de l'interprétation visuelle des images similaires.

5 Bibliographie

[1] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," *Advances in Neural Information Processing Systems*, vol. 6, pp. 737–744, 1993.

[2] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1735–1742, 2006.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[4] Y.-A. Chung and W.-H. Weng, "Learning deep representations of medical images using Siamese CNNs with application to content-based image retrieval," *arXiv*, 2017.

[5] K. Obaideen and M. Al-Shabi, "Siamese networks in medical imaging: A bibliometrics analysis," *DMIP '23: Proceedings of the 2023 6th International Conference on Digital Medicine and Image Processing*, pp. 111–116, 2023.

[6] F. Z. Boroujeni, D. H. Afshari, and F. Mahmoodi, "COVID-19 image retrieval using Siamese deep neural network and hashing technique," *2023 14th International Conference on Information and Knowledge Technology (IKT)*, 2023.

[7] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. Al-Emadi, M. B. I. Reaz, and M. T. Islam, "Can AI help in screening viral and COVID-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132665–132676, 2020.

[8] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, and M. E. H. Chowdhury, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *arXiv*, 2020.

[9] T. Rahman, "COVID-19 Radiography Database," *Kaggle Dataset*. Available: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database. Accessed: May 26, 2026.

[10] G. Huang, Z. Liu, L. Van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[12] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 6105–6114, 2019.

[13] J. P. Cohen, M. Hashir, R. Brooks, and H. Bertrand, "On the limits of cross-domain generalization in automated X-ray prediction," *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, vol. 121, pp. 136–155, 2020.

[14] J. P. Cohen, J. D. Viviano, P. Bertin, P. Morrison, P. Torabian, M. Guarrera, M. P. Lungren, A. Chaudhari, R. Brooks, M. Hashir, and H. Bertrand, "TorchXRayVision: A library of chest X-ray datasets and models," *Proceedings of the 5th Conference on Medical Imaging with Deep Learning*, vol. 172, pp. 231–249, 2022.

[15] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV*, P. S. Heckbert, Ed. Academic Press, pp. 474–485, 1994.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[17] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[18] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[19] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

[21] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[22] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[26] Application de l'apprentissage profond au diagnostic des images médicales : cas d'Alzheimer, *Mémoire de Licence Académique, Spécialité Informatique (SI), Faculté des Sciences, Université M'hamed Bougara – Boumerdès*, 2025.

Les références en couleurs jaune ne sont pas citées dans les différents chapitres