# Rapport de méthodologie

Détection de pneumonie à partir d’images radiographiques (Chest X-ray)  
via Deep Learning et modèles pré‑entraînés

# 1. Problématique

La pneumonie constitue une pathologie respiratoire fréquente et potentiellement grave, en particulier dans les contextes à ressources limitées où l’accès à une expertise radiologique demeure restreint. Le diagnostic repose principalement sur l’interprétation de radiographies thoraciques (Chest X‑ray), un processus exposé à une variabilité inter‑lecteurs et à des difficultés d’interprétation liées à la qualité hétérogène des images et à la similarité visuelle avec d’autres atteintes pulmonaires. L’objectif de ce projet est de proposer une approche d’aide au diagnostic fondée sur l’apprentissage profond, permettant d’identifier automatiquement la présence de pneumonie et de fournir un score de confiance interprétable.

# 2. Objectifs du projet

- Construire un pipeline reproductible de vision par ordinateur pour la classification binaire (Pneumonia vs Normal) sur radiographies thoraciques.

- Exploiter des modèles pré‑entraînés (transfer learning) pour améliorer la performance avec des données annotées limitées.

- Comparer deux familles d’approches : Embeddings + modèles légers (régression logistique / SVM) et Fine‑tuning de CNN pré‑entraînés (ResNet, EfficientNet, DenseNet).

- Conduire une étude d’ablation afin d’identifier les composants déterminants (data augmentation, choix du backbone, profondeur de fine‑tuning).

- Optimiser l’inférence pour contrainte GPU (T4 Colab) via quantization (FP16/INT8) et bonnes pratiques de test‑time.

- Évaluer la solution avec des métriques robustes (AUC, F1‑score, rappel, précision) et une explicabilité (Grad‑CAM) adaptée au cadre médical.

# 3. Données utilisées

Le projet s’appuie sur des jeux de données publics de radiographies thoraciques, notamment le dataset Chest X‑Ray Pneumonia (deux classes : Normal / Pneumonia). Les données feront l’objet d’un contrôle de qualité (résolution, doublons) et d’une préparation standardisée (conversion d’images, harmonisation du format), puis seront partitionnées en jeux d’entraînement, validation et test en respectant, si possible, une séparation par patient pour prévenir les fuites de données.

## 4. Méthodologie prévue

## 4.1. Prétraitement et augmentation

Redimensionnement (224×224 ou 256×256 selon le backbone), normalisation par statistiques ImageNet, et data augmentation contrôlée (flip horizontal, rotations légères ≤ 10°, translations, variations de luminosité/contraste). Ces transformations ne seront appliquées que sur le jeu d’entraînement.

## 4.2. Approche A — Embeddings + modèles légers (baseline CPU)

Extraction d’embeddings via des backbones gelés (ResNet50, EfficientNet‑B0, DenseNet121) en utilisant la couche de Global Average Pooling ; classification par régression logistique ou SVM linéaire. Cette approche sert de référence simple et légère pour quantifier l’apport du fine‑tuning.

## 4.3. Approche B — Transfer learning (fine‑tuning CNN)

Backbones candidats : ResNet50, EfficientNet‑B0/B3, DenseNet121 (poids ImageNet). Tête de classification : Global Average Pooling → Dropout → Couche fully‑connected (1 logit) → Sigmoïde. Schéma d’entraînement : Phase 1, gel complet du backbone et entraînement de la tête ; Phase 2, dégel des derniers blocs avec un taux d’apprentissage réduit et scheduler (Cosine/Step), optimisation AdamW et arrêt anticipé. Utilisation de la précision mixte (AMP) pour réduire l’empreinte mémoire sur T4.

## 4.4. Étude d’ablation (dessin expérimental)

A1 — Data augmentation : sans vs avec ; A2 — Backbone : ResNet50 vs EfficientNet‑B0 vs DenseNet121 ; A3 — Profondeur de fine‑tuning : tête seule vs derniers blocs vs fine‑tuning complet ; A4 — Stratégie de décision : seuil fixe (0,5) vs seuil optimisé sur la courbe Precision‑Recall (maximisation du F1 ou du rappel).

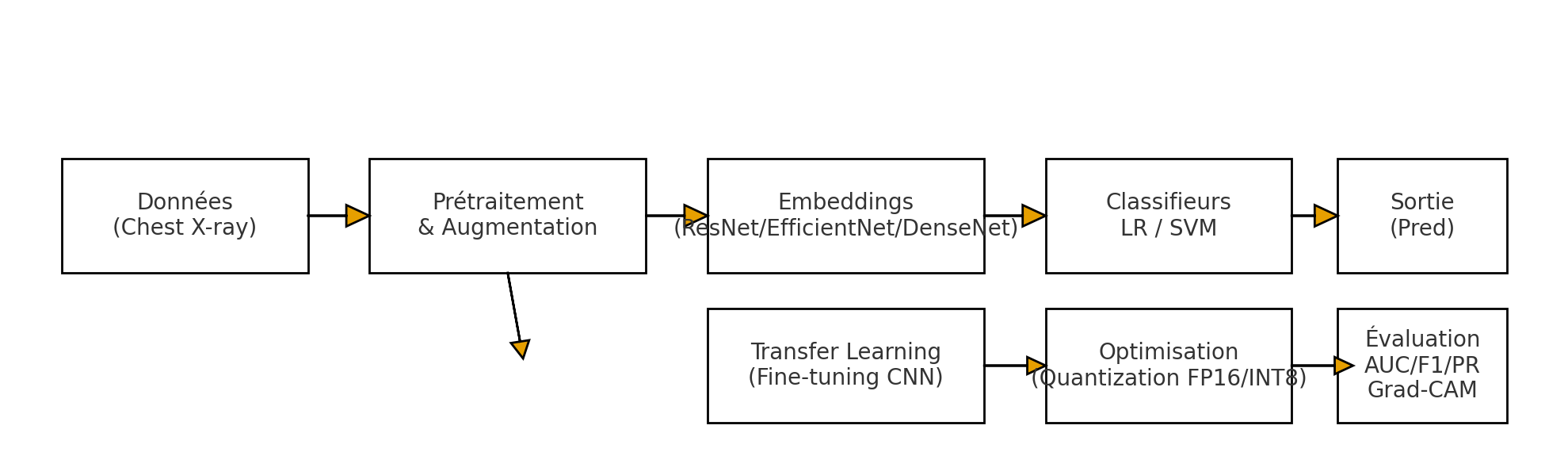
## 4.5. Optimisation pour l’inférence (test‑time)

Quantization FP16/INT8 (ONNX Runtime ou TensorRT) avec calibration sur un sous‑ensemble de validation ; batching raisonnable, fusions de couches, pré‑chauffage, et verrouillage du thread‑pool CPU pour la stabilité. Cible matérielle : GPU T4 (Colab) ; option CPU via la baseline embeddings + classifieurs.

## 4.6. Évaluation et validation

Métriques principales : AUC‑ROC, F1‑score, rappel, précision, accuracy ; courbe Precision‑Recall en cas de déséquilibre de classes. Analyse d’erreurs par matrice de confusion et revue qualitative d’images. Explicabilité par Grad‑CAM afin d’assurer la plausibilité clinique des zones mises en évidence.

## Schéma du pipeline (option O5)



## 5. Conclusion

Le projet propose un système d’aide au diagnostic pour la détection de la pneumonie sur radiographies thoraciques fondé sur des réseaux convolutifs pré‑entraînés. La combinaison d’une baseline par embeddings et de stratégies de fine‑tuning permet une évaluation méthodologiquement rigoureuse, tandis que les optimisations d’inférence (FP16/INT8) garantissent une utilisation efficace des ressources (T4). Les métriques robustes et l’explicabilité (Grad‑CAM) renforcent la crédibilité clinique de l’approche.

# 6. Bibliographie (option O3)

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition (ResNet). CVPR.

- Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML.

- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks (DenseNet). CVPR.

- Selvaraju, R. R., Cogswell, M., et al. (2017). Grad‑CAM: Visual Explanations from Deep Networks via Gradient‑based Localization. ICCV.

- Kaggle. Chest X‑Ray Pneumonia Dataset. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

- ONNX Runtime. Quantization. https://onnxruntime.ai/

- NVIDIA TensorRT. Developer Guide. https://developer.nvidia.com/tensorrt