

# SOUTENANCE DU PROJET S5

*Filière DATA SCIENCE & IoT*

## **Architectures d'apprentissage profond appliquées à la classification du dépistage du cancer du sein**



**Presented by:**

Fadwa LACHAM

Ghizlane BENANI

**Jury Members:**

Pr. IDRI

Pr. ELAOUTAOUI

Pr. ABNANE

# Plan du travail

---

- 1-Introduction
- 2-Présentation du dataset
- 3-Pipeline du prétraitement
- 4-Modélisation et évaluation
- 5-Implémentation
- 6-Conclusion and Perspectives

# 1 INTRODUCTION

# Introduction

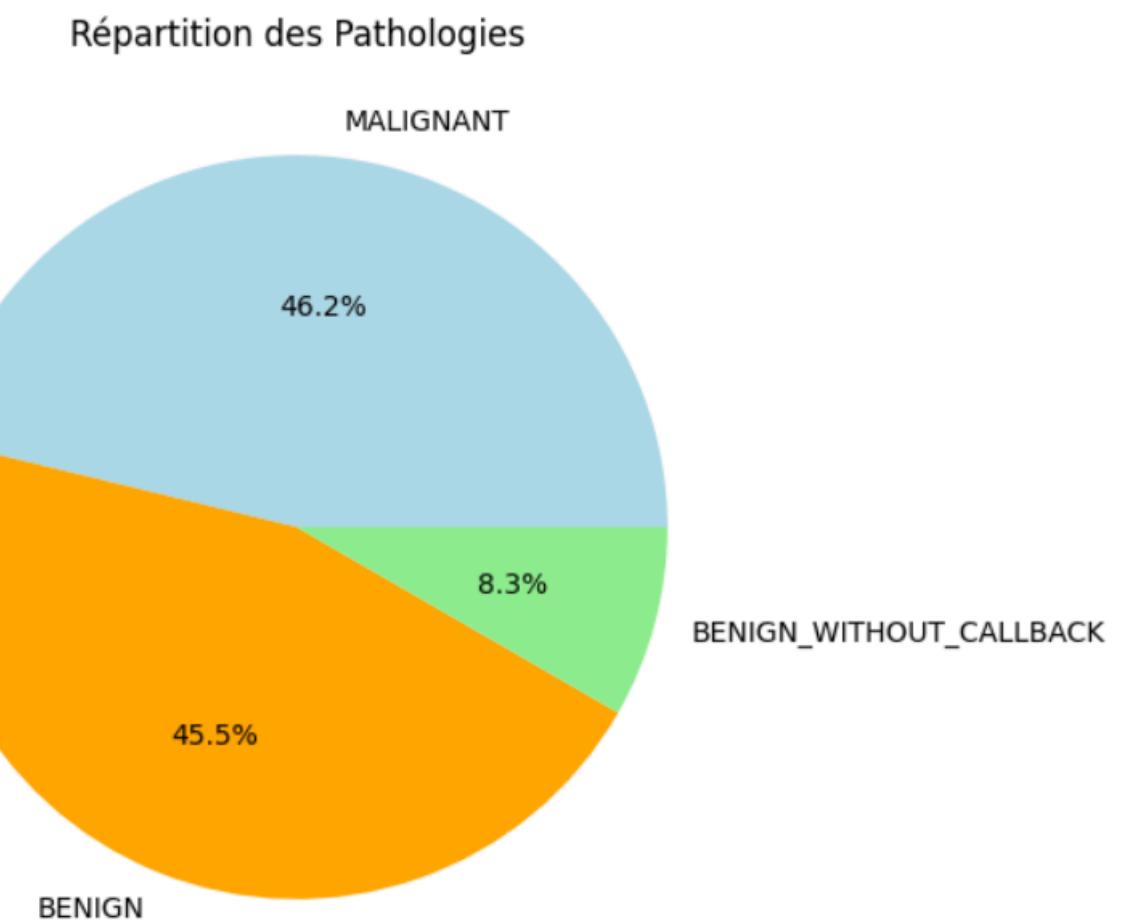
---

- Importance de la détection précoce du cancer du sein.
- Limites des méthodes traditionnelles (erreurs humaines, complexité des données).
- Rôle clé du prétraitement des données dans l'amélioration des performances des modèles CNN.

# 2 Présentation du DataSet

# Description du dataset CBIS-DDSM

- Origine du Dataset :
  - Nom : CBIS-DDSM (Curated Breast Imaging Subset of DDSM).
  - Source : Disponible sur Kaggle .
- Structure des données : Composée de sous-datasets pour les anomalies **calcifications** et **masses**, chacune organisée en ensembles de **train** et **test**.
- Volume des données : Environ 4 000 images mammographiques.
- Classes disponibles : **Bénignes**, **Malignes**, et **Benignes sans suivi**.
- Métadonnées associées : vue de l'image, forme des masses, etc.



# 3 Pipeline du Prétraitement

# Pre Traitement

---

Conversion en  
niveaux de gris

Amélioration  
du contraste

Evaluation de  
l'amélioration du contrast

Normalisation  
des pixels

Redimensionnement  
des images

Équilibrage des  
classes

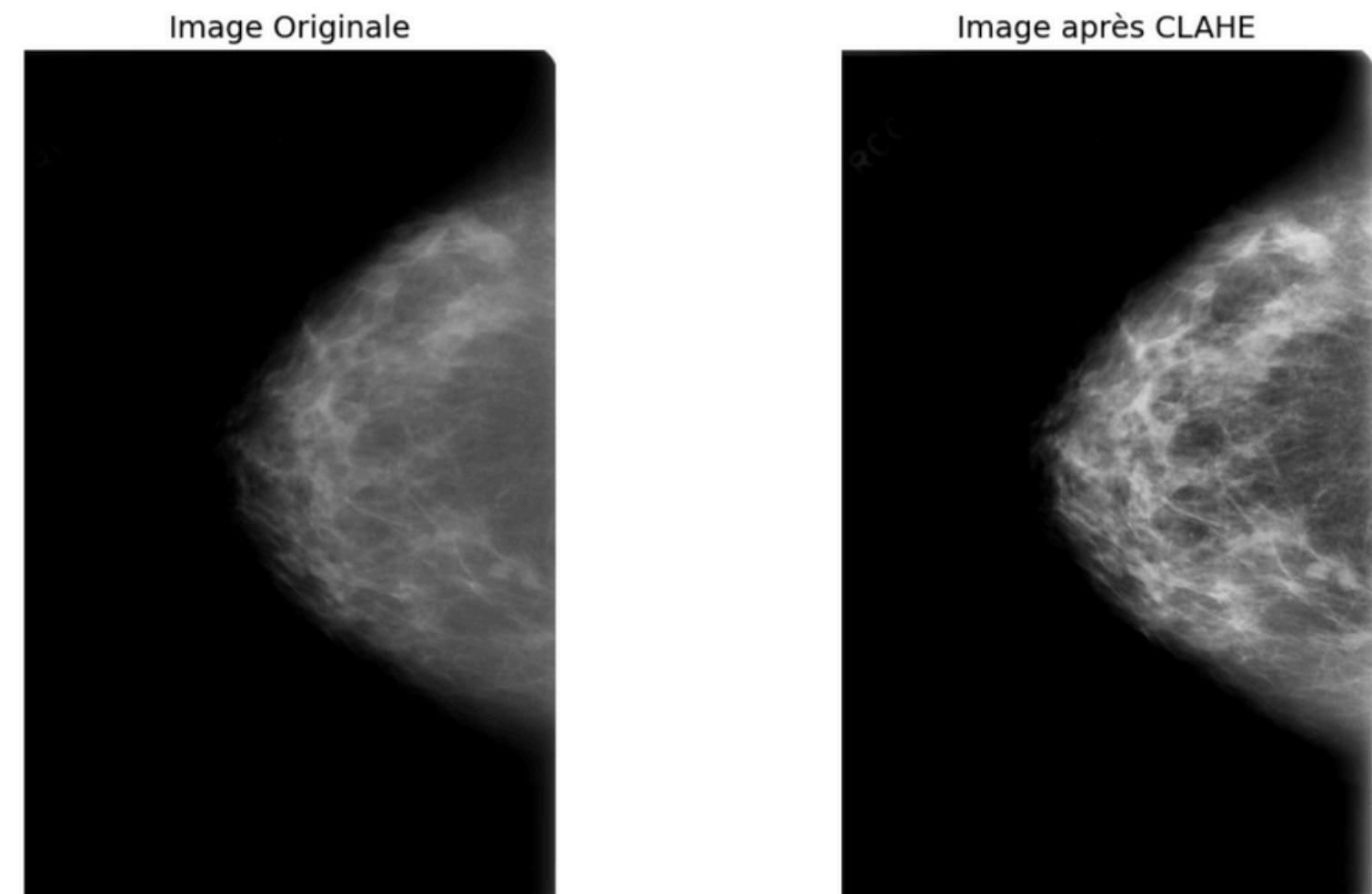
# Amélioration du Contraste

---

- Importance : Améliorer la visibilité des anomalies dans les mammographies.
- Méthode utilisée : CLAHE (Contrast Limited Adaptive Histogram Equalization).
- Impact : Répartition uniforme des intensités des pixels.

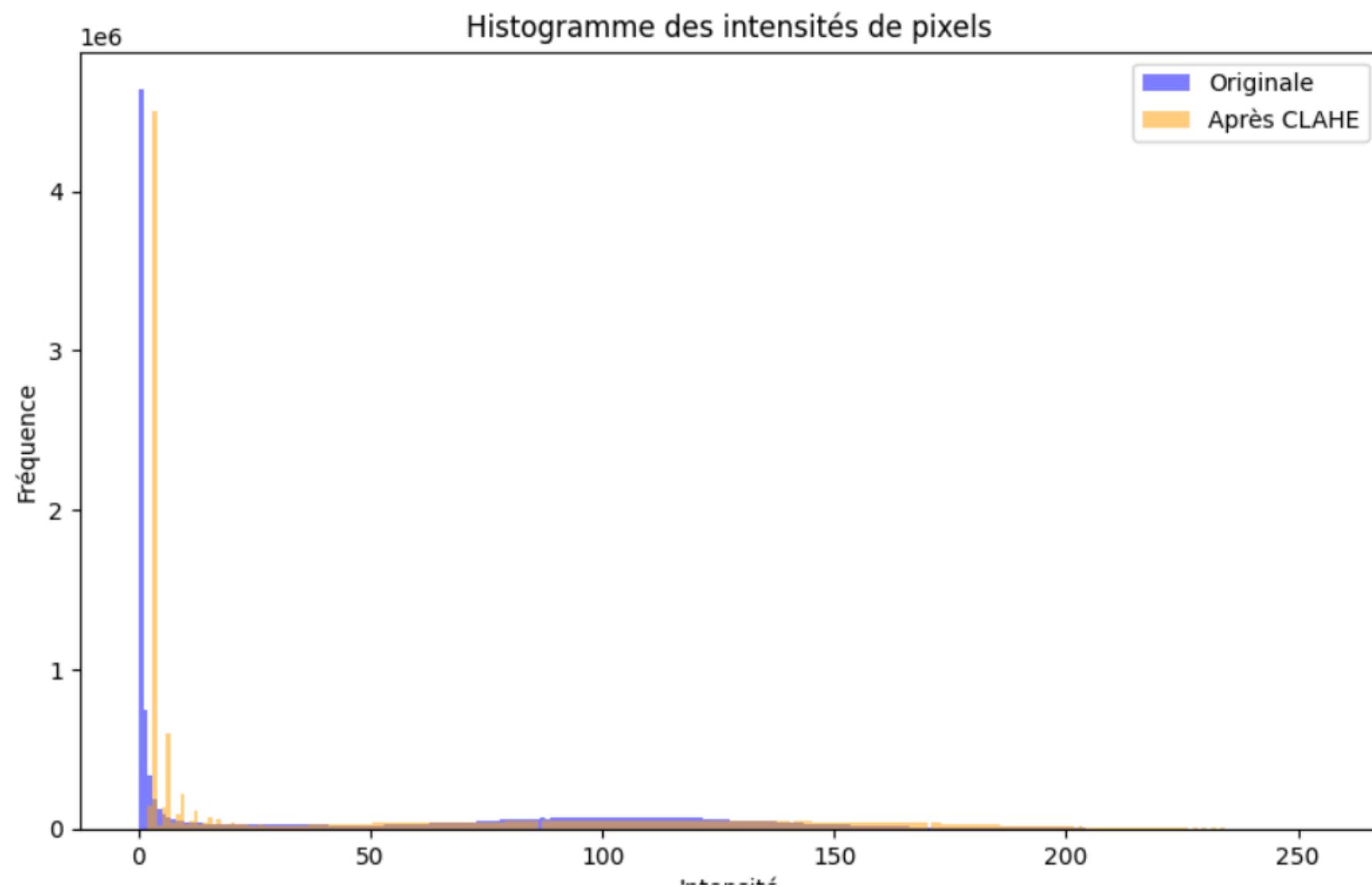
# Impact de CLAHE sur une Mammographie

- **À gauche** : L'image brute montre des niveaux de contraste relativement faibles, ce qui peut rendre certaines anomalies moins visibles.
- **À droite** : L'application de CLAHE améliore la visibilité des structures internes en augmentant localement le contraste, ce qui facilite l'interprétation et la classification par des modèles d'apprentissage profond.



# Evaluation de CLAHE

- Avant CLAHE : Concentration des pixels sur les basses intensités (image sombre).
- Après CLAHE : Répartition plus uniforme des intensités, pic restant bas.

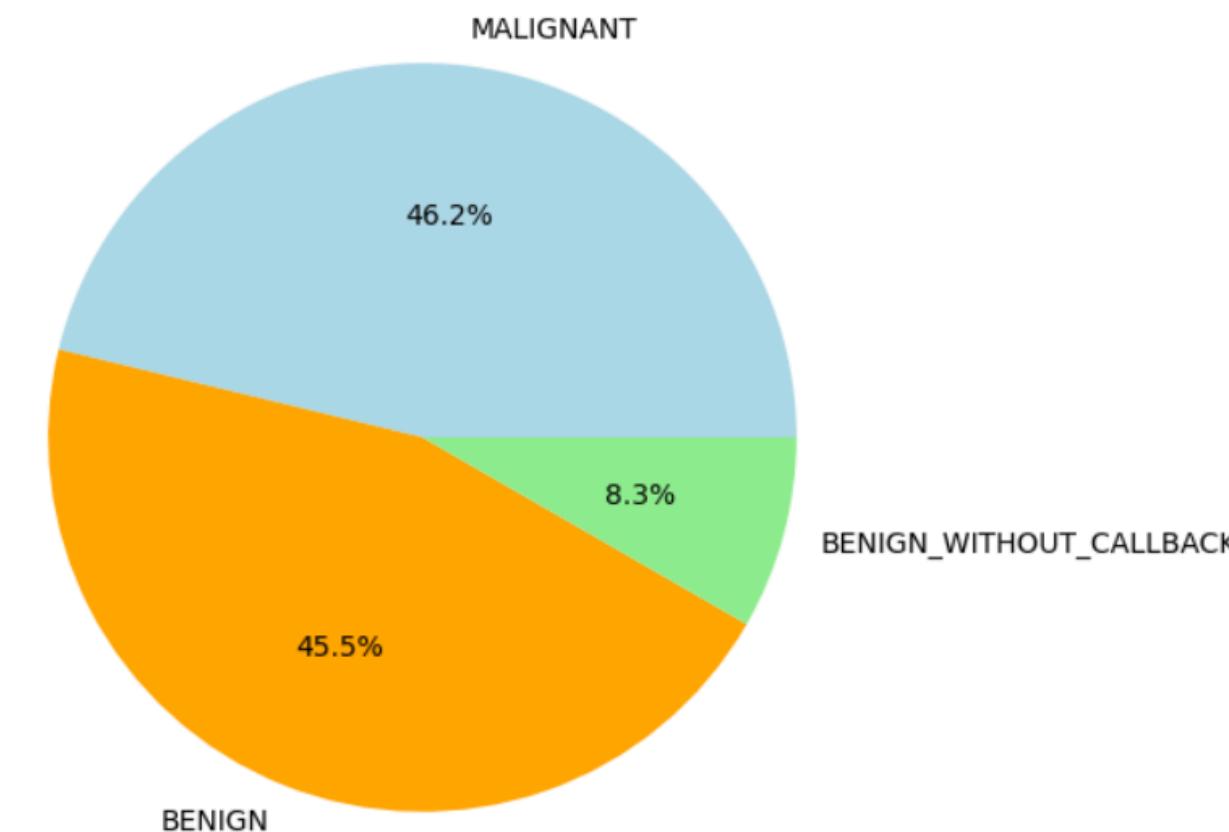
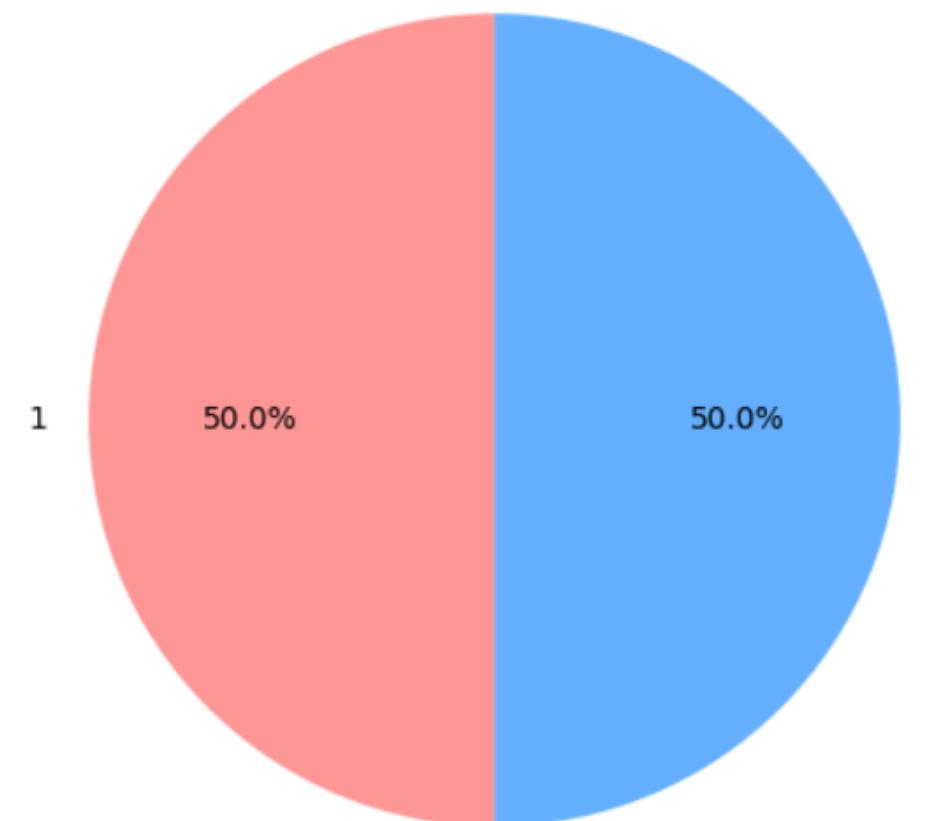


# Normalisation et Redimensionnement

- **Normalisation** : Échelle des pixels ajustée entre 0 et 1 pour accélérer la convergence.
- **Redimensionnement** : Uniformisation des tailles d'images à 224x224x3 pour les rendre compatibles avec les modèles CNN.
- **Résultat** : Réduction des variations non pertinentes dans les données.

# Équilibrage des Classes : Undersampling

- **Problème initial** : Déséquilibre des classes (BENIGN sur-représentée).
- **Solution** : Undersampling pour équilibrer les données.
- **Avantages** : Amélioration des performances des modèles et réduction du biais.
- 'MALIGNANT': 1, 'BENIGN': 1, 'BENIGN\_WITHOUT\_CALLBACK': 0



# Visualisation des Données Prétraitée

Image originale

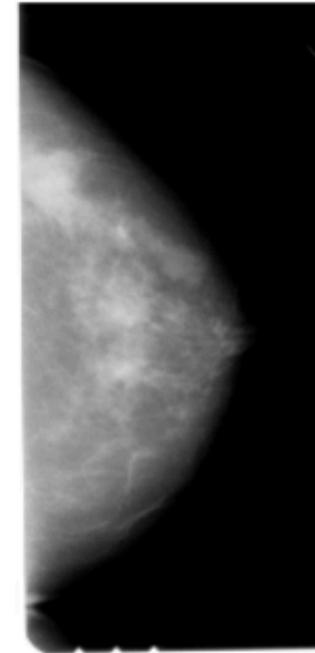


Image après traitement

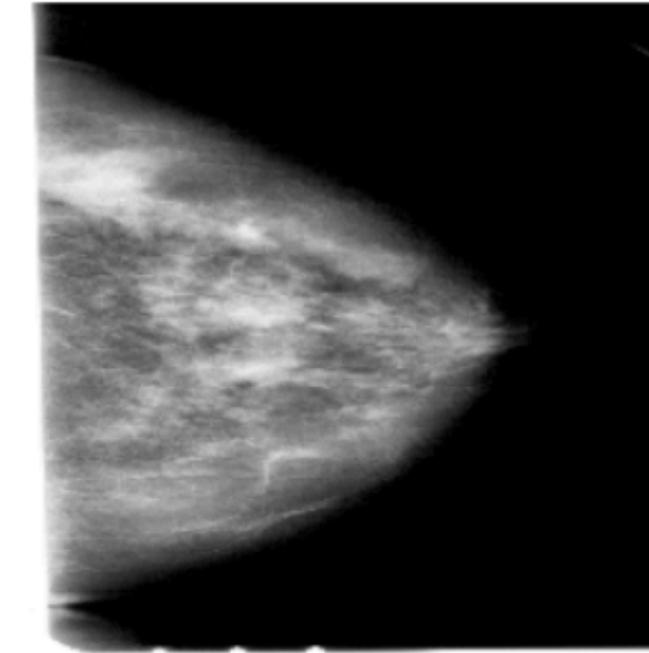


Image originale

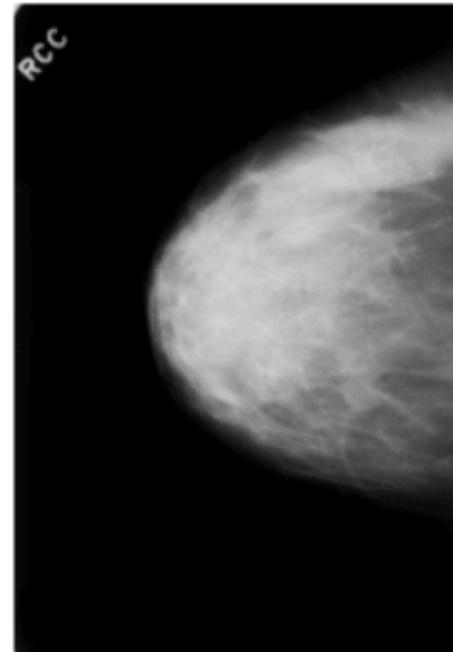
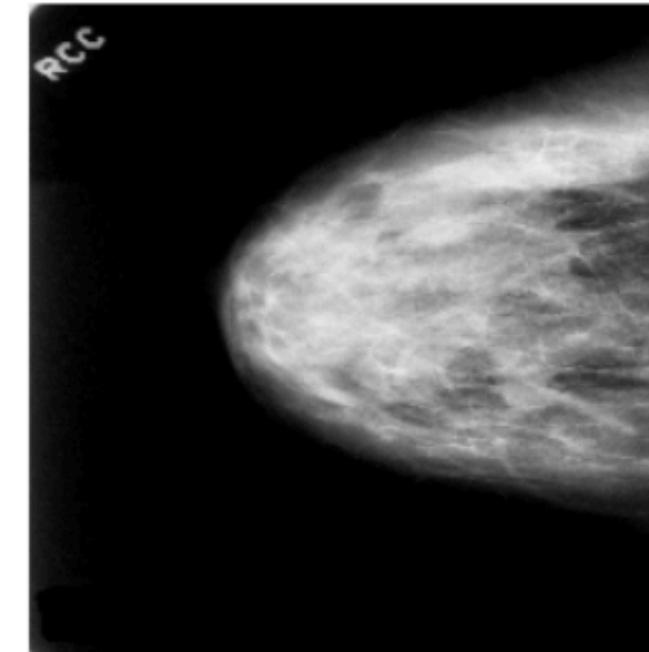


Image après traitement



# 4 Modelisation et évaluation

# Méthodologie

**Dans notre projet, nous avons utilisé trois modèles de deep learning pour atteindre nos objectifs :**

VGG16

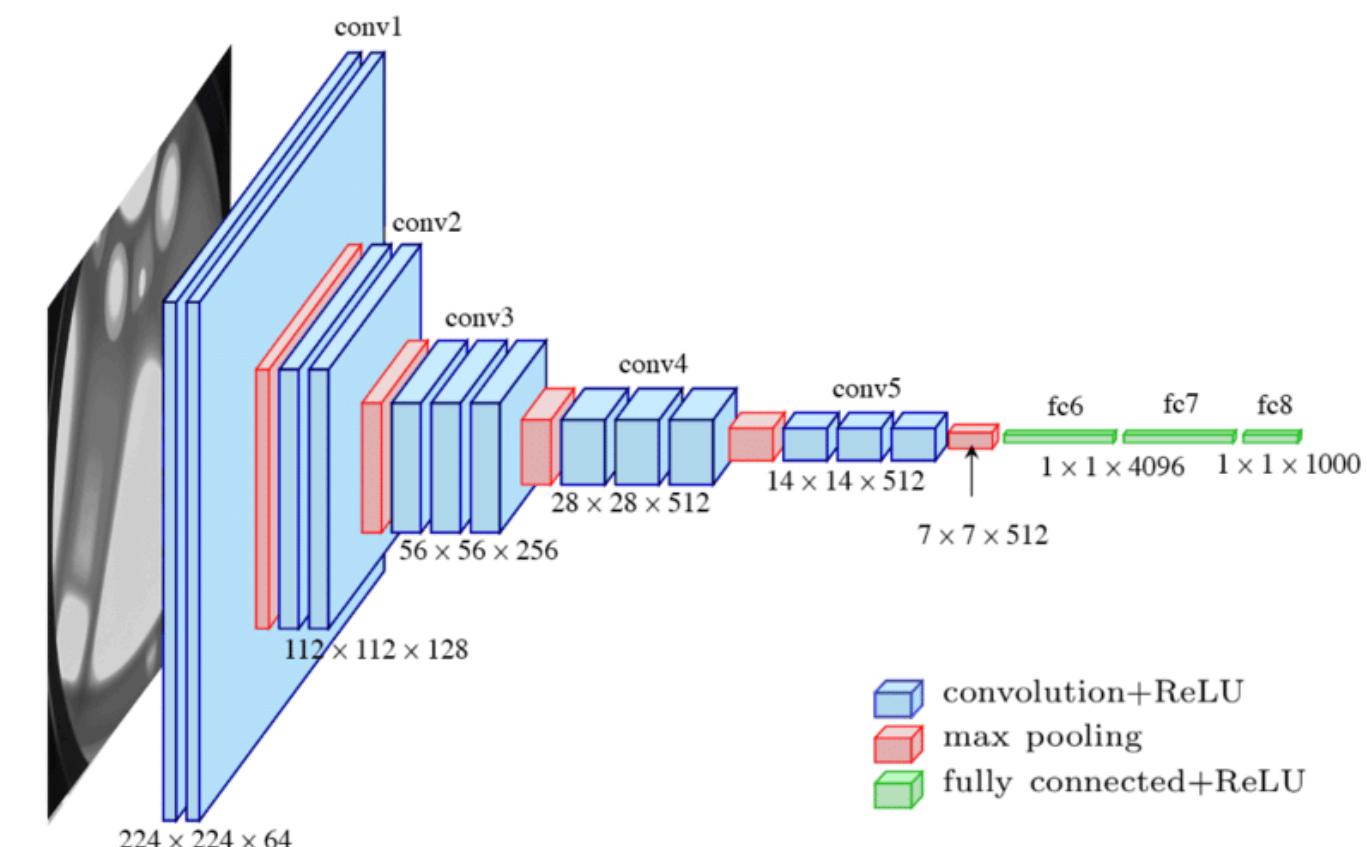
VGG19

DenseNet201

# VGG16 - Architecture et Entraînement

## Architecture VGG16 :

- VGG16 est un réseau de neurones convolutifs composé de 16 couches (d'où le nom), dont 13 couches convolutives et 3 couches entièrement connectées. Ce modèle est reconnu pour sa simplicité et son efficacité dans l'extraction des caractéristiques des images.



# VGG16 - Architecture et Entraînement

---

## Modèle final adapté :

- **Base** : VGG16 pré-entraîné sur ImageNet comme modèle de base, sans les couches de classification finales.
- **Couches ajoutées** : 3 couches Dense de 1024 neurones avec ReLU, et une couche de sortie Sigmoid pour la classification binaire (cancer ou pas de cancer).
- **Régularisation** : Des couches Dropout et BatchNormalization sont ajoutées pour éviter le surajustement et améliorer la convergence.

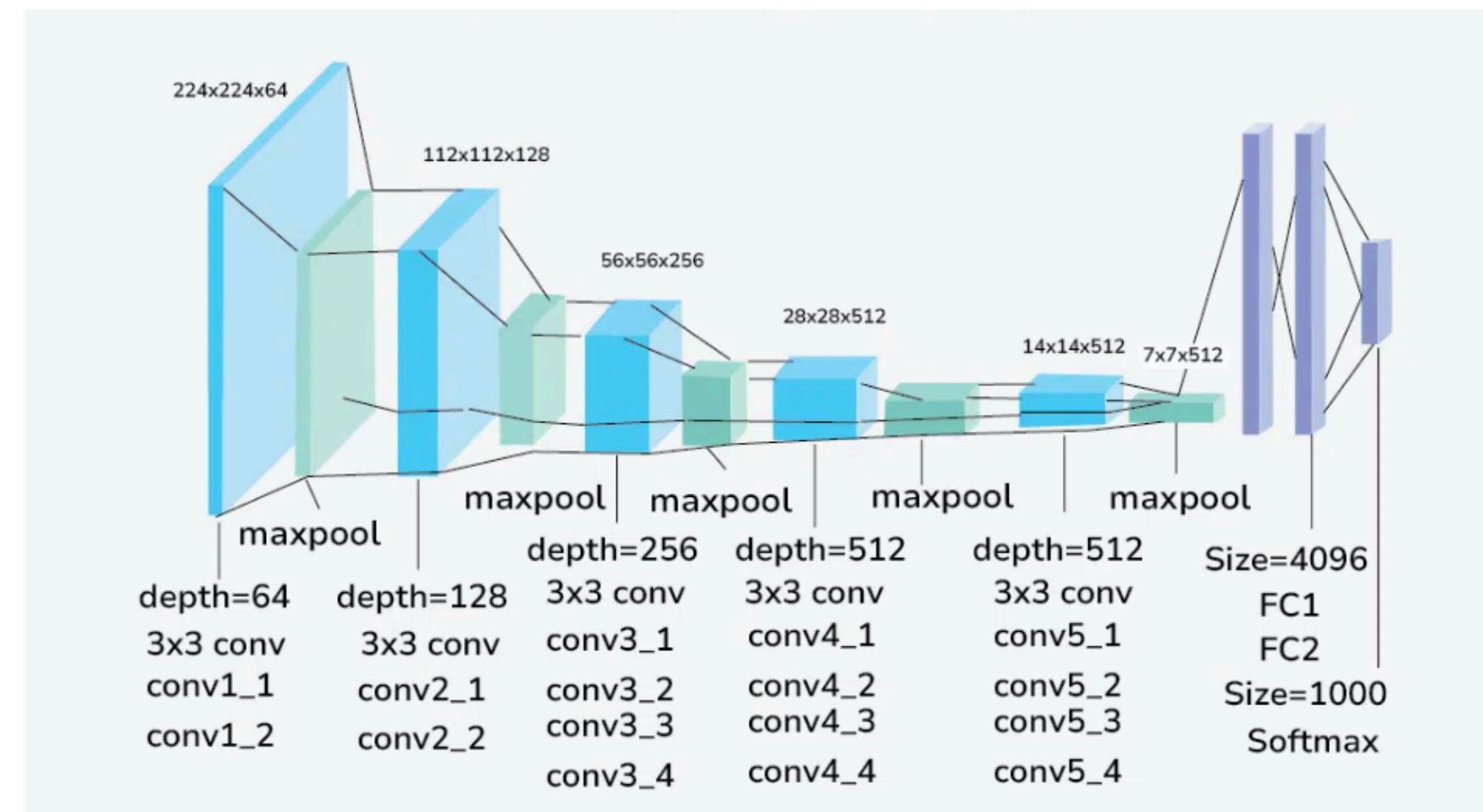
## Entraînement :

- **Époques** : 10 (ajustables selon la performance).
- **Taille du batch** : 32.

# VGG19 - Architecture et Entraînement

## Architecture VGG19 :

- **VGG19 est un réseau de neurones convolutifs profond composé de 19 couches (d'où le nom), dont 16 couches convolutives et 3 couches entièrement connectées. Il est connu pour sa simplicité et son efficacité.**



# VGG19 - Architecture et Entraînement

---

## Modèle final adapté :

- **Base** : Nous avons utilisé VGG19 pré-entraîné sur ImageNet comme modèle de base, sans les couches de sortie (pour éviter l'entraînement des couches de classification initiales).
- **Couches ajoutées** : Des couches Dense supplémentaires, avec Dropout et BatchNormalization pour améliorer la régularisation et éviter le surajustement.
- **Sortie** : Une seule unité de sortie avec activation Sigmoid pour une classification binaire (présence ou absence de cancer).

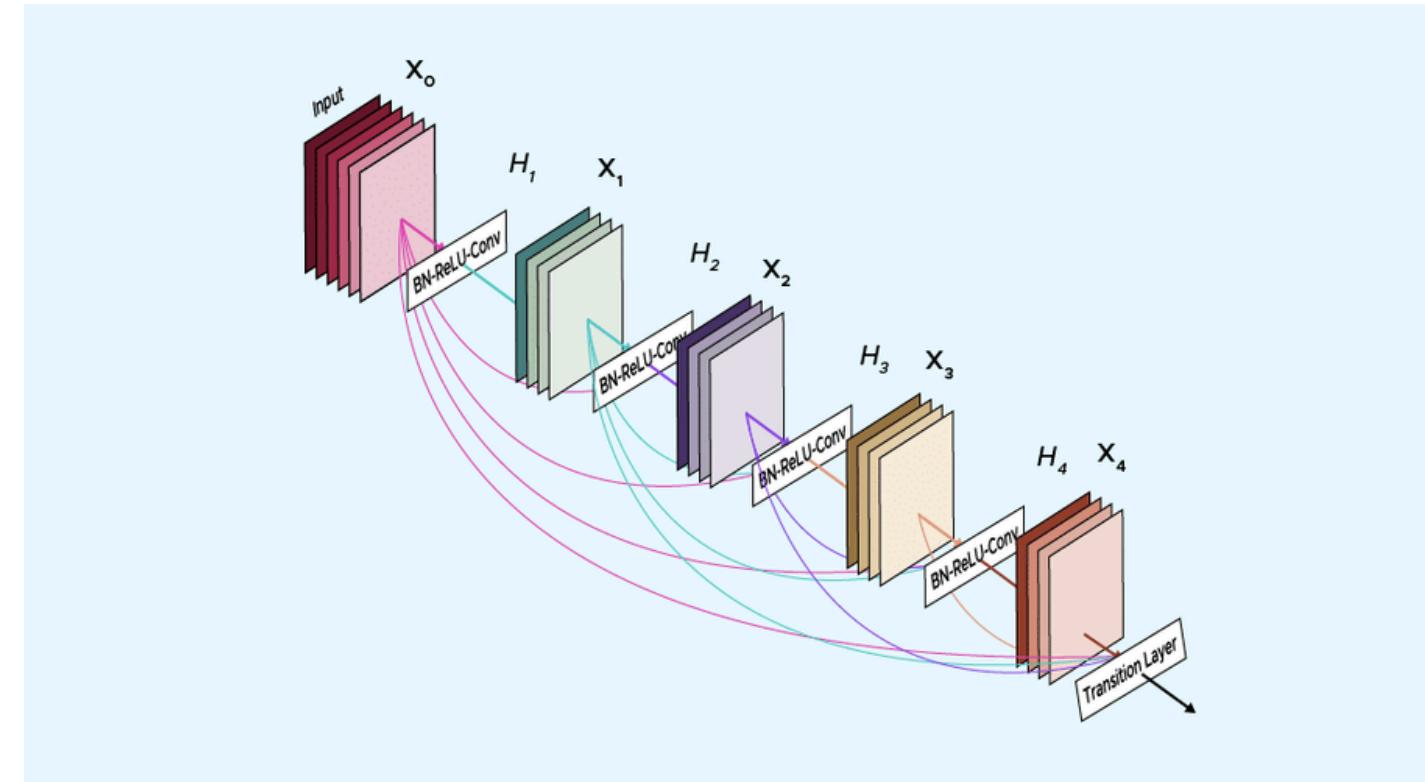
## Entraînement :

- **Époques** : 10.
- **Batch size** : 32.

# DenseNet201 - Architecture et Entraînement

## Architecture DenseNet201 :

- DenseNet201 est un modèle de réseau de neurones convolutifs composé de 201 couches.
- Il utilise une approche d'architecture densément connectée où chaque couche reçoit en entrée toutes les couches précédentes, ce qui améliore l'efficacité de l'entraînement et réduit les problèmes de gradients vanishing.



# DenseNet201 - Architecture et Entraînement

---

## Modèle final adapté :

- **Base** : DenseNet201 pré-entraîné, sans la couche de classification finale.  
Nous avons chargé les poids pré-entraînés à partir du chemin local.
- **Couches ajoutées** : 3 couches Dense de 1024 neurones avec ReLU, et une couche de sortie Sigmoid pour la classification binaire (cancer ou pas de cancer).
- **Régularisation** : Des couches Dropout et BatchNormalization pour éviter le surajustement et accélérer l'entraînement

# Évaluation des Performances des Modèles

## Métriques d'évaluation :

Accuracy

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{true negatives} + \text{false negatives} + \text{false positives}}$$

Precision

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

F1 Score

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Recall

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

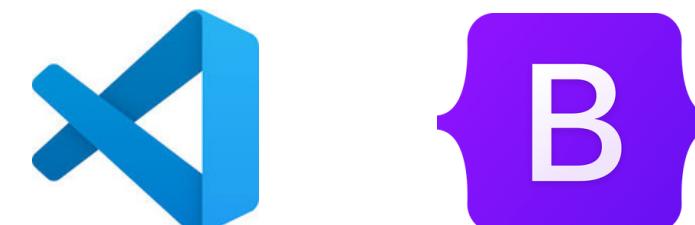
# Évaluation des Performances des Modèles

## Tableau Comparatif :

Modèle	Précision (Classe 0)	Rappel (Classe 0)	F1-score (Classe 0)	Précision (Classe 1)	Rappel (Classe 1)	F1-score (Classe 1)	Accuracy	Macro Avg	Weighted Avg
VGG16	0.69	1.00	0.81	1.00	0.52	0.69	0.77	0.84	0.84
VGG19	0.61	1.00	0.76	1.00	0.33	0.50	0.67	0.81	0.80
DenseNet201	0.77	0.91	0.83	0.88	0.71	0.79	0.81	0.83	0.82

# 5 INTERFACE GRAPHIQUE

# Architecture de Projet et Outils



- **Interface Frontend :**

- Développée en utilisant HTML, CSS et JavaScript.
- Reliée au modèle de machine learning via le backend Flask.

- **Backend :**

- Application Flask servant les templates HTML.
- Héberge le modèle de machine learning pour le dépistage du cancer du sein

# DÉMONSTRATION

```
Windows PowerShell x + - X
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the
model.
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 747-953-620
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:51] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /static/css/bootstrap.css HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/arrival-bg.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p1.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p2.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /js/jquery-3.4.1.min.js HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /js/popper.min.js HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /static/images/log.png HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /js/bootstrap.js HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /static/css/style.css HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /static/css/responsive.css HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /js/custom.js HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /static/css/font-awesome.min.css HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /static/images/background.jpg HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p3.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p4.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p5.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p6.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p7.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p8.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p9.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p10.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p11.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/p12.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/client.jpg HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:54] "GET /images/logo.png HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [15/Jan/2025 20:41:55] "GET /static/fonts/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1" 304 -
(venv) PS D:\S5\Projet S5\flask_model_app> python app.py
2025-01-15 20:42:37.293953: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to flo
ating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-01-15 20:42:38.101329: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to flo
ating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-01-15 20:42:39.260221: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
```

# 6 Conclusion Perspectives

# Conclusion

---

- Ce projet a permis d'explorer l'utilisation des réseaux de neurones convolutifs (CNN) pour le dépistage précoce du cancer du sein.
- En évaluant les architectures VGG16, VGG19 et DenseNet201, nous avons montré leur potentiel dans l'amélioration de la précision et de la fiabilité des diagnostics à partir des mammographies.

# Perspectives

---

- Intégration de nouvelles sources de données pour améliorer la précision et la fiabilité des modèles.
- Amélioration continue des modèles prédictifs pour s'adapter aux tendances et évolutions du domaine médical.
- Enrichissement de l'interface utilisateur avec des fonctionnalités supplémentaires pour répondre aux besoins évolutifs des radiologues.

MERCI POUR  
VOTRE  
ATTENTION