**BreastScreen-AI :**

**Un outil de détection précoce du cancer du sein à Djibouti par l’Intelligence Artificielle**

**Raffinement du modèle**

**Membres de l’équipe : Yacin MOUHOUMED ELMI**

1. **Aperçu**

La phase de raffinement du modèle a pour objectif d’améliorer les performances initiales obtenues avec le CNN. Elle repose sur l’évaluation des résultats initiaux, l’identification des faiblesses (notamment au niveau du rappel pour certaines classes), et l’application de techniques de régularisation, d’optimisation et de réglage d’hyperparamètres pour atteindre une meilleure généralisation sur les données de validation et de test.

1. **Évaluation du modèle**

Après l’entraînement initial du CNN sur 50 époques, le modèle a atteint :

* **Précision sur validation : 81,56 %**
* **Perte sur validation : 2,2485**

L’évaluation sur l’ensemble de test a montré :

* **Précision sur test : 85,00 %**
* **Perte sur test : 2,5585**

Le rapport de classification (validation) révèle :

* Classe **Normal** : précision = 0.74, rappel = 0.86, f1-score = 0.80
* Classe **Benign** : précision = 0.87, rappel = 0.85, f1-score = 0.86
* Classe **Malignant** : précision = 0.74, rappel = 0.70, f1-score = 0.72

Le modèle présente une bonne performance générale, mais une **faiblesse de rappel pour la classe "Malignant"**, qui est critique en dépistage médical.

1. **Techniques de raffinement**

Pour améliorer les performances, plusieurs stratégies ont été adoptées :

* **Dropout** : déjà appliqué (0.1) pour limiter le surapprentissage.
* **Data augmentation** : prévue pour enrichir le jeu d’entraînement (rotations, zooms, flips horizontaux).
* **Optimisation** : Adam a été retenu, mais d’autres optimisateurs (RMSprop, SGD avec momentum) seront testés.

1. **Réglage des hyperparamètres**

Des ajustements ont été envisagés :

* Taille du batch : **32** (tests à faire avec 16 et 64 pour comparer la convergence).
* Nombre d’époques : **50** (augmentation possible à 100 avec *early stopping*).
* Taux d’apprentissage : ajustement fin via *LearningRateScheduler*.

Ces ajustements permettront de stabiliser la courbe de perte validation et d’éviter les oscillations observées.

1. **Validation croisée**

La validation croisée n’a pas été mise en place de manière exhaustive en raison du coût computationnel. Cependant, une stratégie de **stratified k-fold cross-validation** est prévue afin d’assurer une meilleure représentativité des classes minoritaires (notamment les cas malins).

1. **Sélection des caractéristiques**

Le CNN apprend automatiquement les caractéristiques à partir des images échographiques, donc aucune ingénierie manuelle n’a été nécessaire. Cependant, une sélection indirecte a été réalisée par normalisation des images et resizing en 256x256 pixels.

**Soumission de Test**

1. **Aperçu**

Le modèle final a été appliqué sur un **jeu de test indépendant (x\_test\_resized)** pour vérifier sa capacité de généralisation.

1. **Préparation des données pour les tests**

Les données de test ont été redimensionnées (256x256), normalisées, et séparées des ensembles train/validation afin d’assurer une évaluation impartiale.

1. **Application du Modèle**

# Évaluation directe du modèle sur le test set

test\_loss, test\_accuracy = model.evaluate(x\_test\_resized, y\_test, verbose=0)

print(f"📊 Perte sur test : {test\_loss:.4f}")

print(f"✅ Précision sur test : {test\_accuracy\*100:.2f}%")

1. **Métriques de Test**

* **Précision sur test : 85,00 %**
* **Perte sur test : 2,5585**

📋 Rapport de classification (test) :

* Normal : précision 0.75, rappel 0.83
* Benign : précision 0.88, rappel 0.86
* Malignant : précision 0.73, rappel 0.71

Le modèle reste performant, mais nécessite un raffinement spécifique sur la classe *Malignant* afin de réduire les faux négatifs.

1. **Déploiement du Modèle**

Le modèle a été sauvegardé en format .h5 et réutilisé pour des prédictions unitaires. Il sera déployé via **Gradio App sur Hugging Face Spaces**, offrant une interface conviviale et un accès sécurisé aux médecins pour le dépistage assisté par IA.

1. **Implémentation du Code**

Exemple : courbes d’apprentissage (accuracy & loss).

plt.figure(figsize=(12, 4))

# Accuracy

plt.subplot(1, 2, 1)

plt.plot(hist.history['accuracy'], label='Entraînement')

plt.plot(hist.history['val\_accuracy'], label='Validation')

plt.title("🎯 Précision (accuracy)")

plt.xlabel("Époque")

plt.ylabel("Précision")

plt.legend()

# Loss

plt.subplot(1, 2, 2)

plt.plot(hist.history['loss'], label='Entraînement')

plt.plot(hist.history['val\_loss'], label='Validation')

plt.title("📉 Perte (loss)")

plt.xlabel("Époque")

plt.ylabel("Perte")

plt.legend()

plt.tight\_layout()

plt.show()

**Conclusion**

Le modèle CNN a atteint **85 % de précision sur le jeu de test**, démontrant son potentiel pour le dépistage automatique du cancer du sein à partir d’échographies. Toutefois, la détection des cas **malins** doit être renforcée. Le raffinement passera par l’**augmentation des données** et l’**optimisation des hyperparamètres**. Le déploiement sur **Gradio (Hugging Face)** rendra ce modèle accessible et utilisable par les professionnels de santé.

**Références**

* TensorFlow & Keras Documentation.
* BUSI Dataset (2020). Breast Ultrasound Images Dataset - Open Access.
* Mohamed, I. et al. (2023). Lightweight CNN for breast cancer detection. arXiv.
* Zhang, Q. et al. (2021). EfficientNet-based automated breast cancer diagnosis. IEEE Access.