FACULTÉ DES SCIENCES DHAR EL MAHRAZ
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH

Université SIDI MOHAMED BEN ABDELLAH

Segmentation de Tumeurs Breast&Brain

Présenté par :

- EL MAHDAOUI Ahmed
- ER-ROUGBANI Mouaad
- LAHMAMSSI Adnane

Encadré par :
Dr. RIFFI Jamal

Année universitaire 2024–2025

Contents

1. Introduction

La détection et la segmentation automatiques des tumeurs dans les images médicales jouent un rôle crucial dans le diagnostic et la planification thérapeutique. Les méthodes classiques, souvent manuelles, sont longues, coûteuses et sensibles aux erreurs humaines.

Grâce aux progrès de l'apprentissage profond, en particulier des réseaux de neurones convolutifs (CNN), il est aujourd'hui possible d'entraîner des modèles capables d'identifier et segmenter efficacement les structures anormales dans les images médicales. Ce rapport présente une étude expérimentale de la segmentation de tumeurs dans deux types d'images : les IRM cérébrales et les échographies mammaires.

2. Jeux de Données Utilisés

2.1. Brain Tumor Dataset

- **Source :** Kaggle
- **Contenu :** Images IRM cérébrales (formats PNG et JPG) avec leurs masques binaires annotés.
- **Structure :** Deux dossiers : `images/` et `masks/`, chaque image a un masque correspondant.

2.2. Breast Ultrasound Images Dataset

- **Source :** Kaggle
- **Contenu :** Échographies mammaires classées (normal, bénin, malin), accompagnées de masques pour les cas tumoraux.
- **Défis :** Bruit visuel élevé, faible contraste, formes de tumeurs variées.

3. Méthodologie

3.1. Prétraitement et Augmentations

- Redimensionnement des images à 512×512.
- Conversion des images en 3 canaux (grayscale $\rightarrow$ RGB).
- Augmentations : rotation, flips, zoom, translation.

3.2. Architecture du Modèle

Nous avons utilisé l'architecture **UNet++** fournie par la bibliothèque `segmentation_models.pytorch`, avec un encodeur **ResNet34** préentraîné sur ImageNet. Ce choix combine les avantages :

- Résolution fine des détails (UNet++)

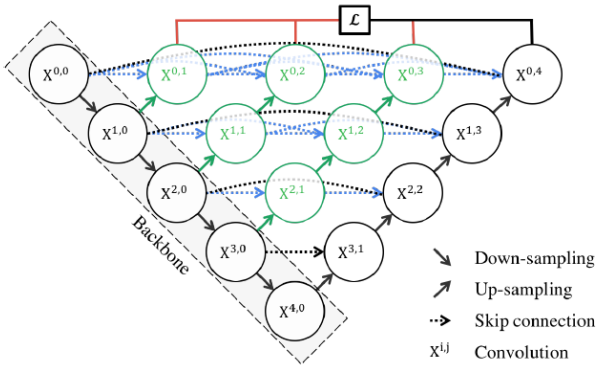

Figure 1: Architecture de U-Net++

- Extraction efficace de caractéristiques (ResNet34)

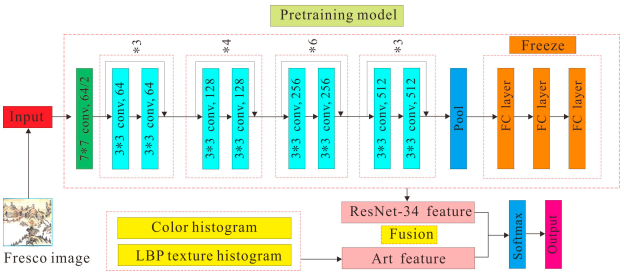

Figure 2: Architecture de ResNet34

3.3. Entraînement

- **Fonction de perte :** Dice Loss + Binary Cross Entropy.

  La fonction de perte hybride combine les avantages du *Dice Loss* (bonne performance sur les classes déséquilibrées) et de la *Binary Cross-Entropy* (optimisation stable) :

$$\mathcal{L}_{\text{Total}} = \underbrace{1 - \frac{2|P \cap G| + \epsilon}{|P| + |G| + \epsilon}}_{\text{Dice Loss}} + \lambda \underbrace{\left[-\frac{1}{N} \sum_{i=1}^{N} g_i \log(p_i) \right]}_{\text{Binary Cross-Entropy}} \tag{1}$$

  – P : Masque prédit (probabilités)

  – G : Masque de vérité terrain (valeurs binaires)

- λ : Poids de la BCE (typiquement $\lambda = 1$)
- ϵ : Terme de lissage (10^{-6})

- **Optimiseur :** Adam, LR $= 1 \times 10^{-4}$

  L'optimiseur Adam combine les avantages de RMSProp et du momentum. Pour chaque paramètre θ_t à l'étape t :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \qquad \text{(Momentum du gradient)} \tag{2}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \qquad \text{Mise à jour des carrés} \tag{3}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \qquad \text{Correction du biais} \tag{4}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \qquad \text{Correction du biais} \tag{5}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \qquad \text{Mise à jour des paramètres} \tag{6}$$

  - g_t : Gradient à l'étape t
  - η : Taux d'apprentissage initial (typiquement 10^{-3})
  - β_1, β_2 : Coefficients de décroissance (par défaut 0.9 et 0.999)
  - ϵ : Terme de régularisation (typiquement 10^{-8})

- **Batch size :** 8
- **Époques :** 25
- **Répartition :**
  - Entraînement : 70%
  - Validation : 20%
  - Test : 10%

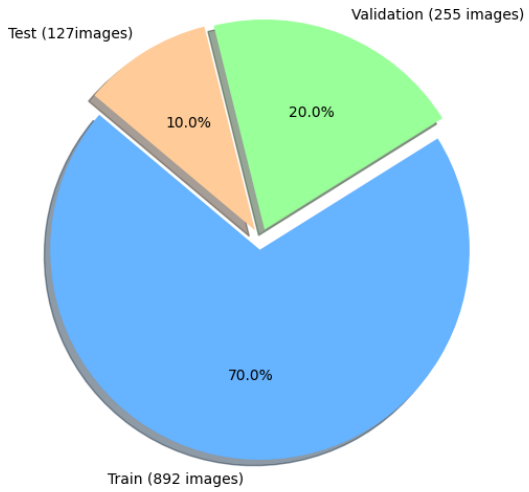

Figure 3: Répartition de données

4. Résultats

4.1. Métriques

Nous avons utilisé principalement le **Dice Coefficient** pour mesurer l'overlap entre le masque prédit et le masque réel. La fonction de perte combinée permet d'équilibrer la précision et la stabilité pendant l'entraînement.

4.2. Performances Observées

```
accuracy, f1, precision, recall = evaluate_model_metrics(model, test_loader)
```

```
Accuracy  : 0.9754
F1 Score  : 0.7828
Precision : 0.8079
Recall    : 0.7592
```

Figure 4: Performance

4.3. Visualisation

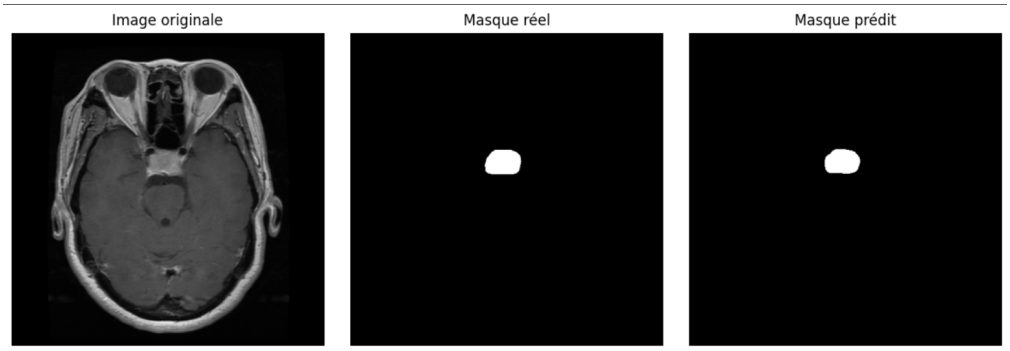

Figure 5: Exemple de segmentation

5. Comparaison des Architectures : U-Net vs U-Net++ vs DeepLabV3

Pour analyser l'impact de l'architecture du modèle sur la segmentation, nous avons comparé trois variantes populaires : **U-Net**, **U-Net++** et **DeepLabV3**, toutes avec le même backbone ResNet34. L'entraînement a été réalisé dans les mêmes conditions pour garantir une évaluation équitable.

Accuracy	: 0.6434
F1 Score	: 0.2410
Precision	: 0.1373
Recall	: 0.9878

: U-Net

Accuracy	: 0.9754
F1 Score	: 0.7828
Precision	: 0.8079
Recall	: 0.7592

: U-Net++

Accuracy	: 0.9747
F1 Score	: 0.7710
Precision	: 0.8099
Recall	: 0.7356

: DeepLabV3

Figure 6: Comparaison des performances des trois architectures de segmentation

Analyse comparative :

- **U-Net++** démontre les meilleures performances globales :
  - Meilleur compromis précision/rappel (F1-Score : 0.7828)
  - Précision élevée (0.8079) tout en conservant un bon rappel (0.7592)
  - Excellente exactitude (0.9754)
  - Particulièrement efficace sur les contours fins grâce à ses connexions imbriquées
- **DeepLabV3** montre des résultats intéressants :
  - Précision légèrement supérieure (0.8099)

  - Performance légèrement inférieure sur le rappel (0.7356) et F1-Score (0.7710)
  - Excellente contextualisation des grandes régions
  - Temps d'entraînement plus long que les autres architectures

- **U-Net** reste compétitif :
  - Rappel exceptionnel (0.9878) mais faible précision (0.1373)
  - F1-Score limité (0.2410) dû au déséquilibre précision/rappel
  - Architecture légère, adaptée aux systèmes embarqués
  - Solution intéressante pour les applications temps réel

Conclusion : U-Net++ se positionne comme le meilleur compromis pour la plupart des applications, tandis que U-Net peut être privilégié lorsque le rappel est critique et que les ressources sont limitées. DeepLabV3 offre une alternative intéressante pour les problèmes nécessitant une analyse contextuelle approfondie.

6. Conclusion

Cette étude a systématiquement évalué différentes architectures de segmentation pour les tumeurs cérébrales et mammaires, établissant que U-Net++ offre le meilleur compromis performance/précision. Bien que des défis persistent (segmentation des petites lésions, robustesse aux artefacts), les résultats démontrent le potentiel clinique de ces approches. Les futures recherches devraient intégrer des mécanismes d'attention et explorer l'apprentissage semi-supervisé pour réduire la dépendance aux annotations manuelles. Ce travail constitue une étape significative vers l'intégration de l'IA dans les workflows diagnostiques en oncologie.

Références

- Dataset Cerveau : `https://www.kaggle.com/datasets/tinashri/brain-tumor-dataset-include`
- Dataset Sein : `https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dat`