

# Apprentissage pour l'image i3D

## Rapport du Projet kaggle

## Introduction

Dans le cadre de L'Unité d'enseignement API (APPRENTISSAGE POUR L'IMAGE), nous devons classer des images 2D issues de microscopie FIB/SEM (FIB : Focused Ion Beam, SEM : Scanning Electron Microscopy) selon 5 classes biologiques :

1. : « autre » (le pixel central n'appartient à aucune classe annotée)
2. : mitochondrie
3. : réticulum endoplasmique
4. : membrane nucléaire
5. : membrane cellulaire

Toutes les informations sont disponibles au challenge kaggle suivant : <https://www.kaggle.com/competitions/2024-competition-asi-i-3-d/overview>.

L'objectif est d'obtenir un modèle performant qui attribue la bonne classe à chaque image. Étant donné la difficulté de collecter et d'annoter ces données de très haute résolution, nous avons choisi de mettre en place des techniques de transfer learning et d'augmentation afin d'atteindre un haut niveau d'accuracy (environ 95%). Nos travaux sont disponibles sur le Notebook suivant : <https://www.kaggle.com/code/florenthardy/notebookbe1e13c73f>.

## Données et Prétraitement

Les données sont constituées de 14 000 images d'entraînement (train) et de 6 000 images de test (test). Chaque image est associée à une classe à prédire.

### Prétraitement des images

- Chargement des données : les fichiers images sont chargés et associés à leur classe grâce à un DataFrame.
- Normalisation : les images sont converties en niveaux de gris et leurs valeurs sont normalisées entre 0.0 et 1.0.
- Encodage des labels : les classes sont converties en format one-hot pour être compatibles avec le modèle.

---

**Algorithm 1:** Prétraitement des images

---

**Input:** *filepath*, *label*, *augment=False*  
**Output:** Image prétraitée et encodée  
**begin**  
    Charger l'image ;  
    Redimensionner l'image à  $256 \times 256$  ;  
    Appliquer une augmentation si *augment=True* ;  
    Normaliser les valeurs entre 0 et 1 ;  
    Encoder le label en one-hot ;  
    **Retourner** l'image et le label encodé  
**end**

---

## Méthodologie

### Augmentation des données

L'augmentation permet de générer artificiellement plus de données pour améliorer la robustesse du modèle. Nous avons appliqué :

- Rotation aléatoire ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ )
- Zoom aléatoire entre 0.8 et 1.0

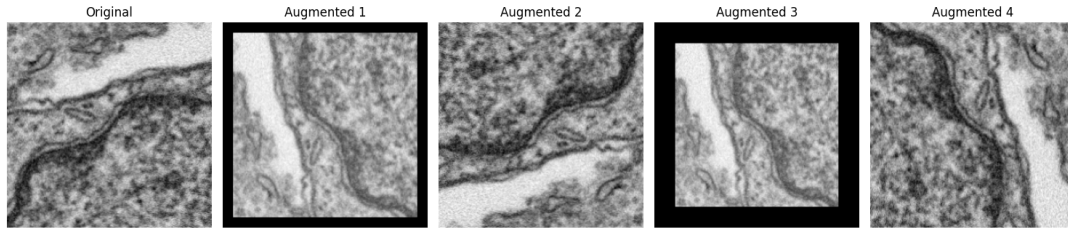


FIGURE 1 – Exemple d’images après augmentation.

## Modèle : EfficientNetB4

Nous avons utilisé le modèle EfficientNetB4 pour le transfert d’apprentissage. Ce modèle est préentraîné sur le dataset ImageNet et offre un bon compromis entre performance et consommation mémoire.

Pour notre tâche, nous avons adapté EfficientNetB4 en ajoutant des couches spécifiques pour répondre aux besoins de classification des 5 classes biologiques :

- **Couches ajoutées :**
  - Une couche *Global Average Pooling* pour réduire les dimensions spatiales de la sortie du réseau de base tout en conservant les informations pertinentes.
  - Une couche *Dense* avec 256 neurones et une activation *ReLU*, pour capturer des caractéristiques complexes spécifiques à la classification.
  - Une couche de *Dropout* avec un taux de 0.5, pour prévenir le surapprentissage (overfitting).
  - Une couche *Dense* finale avec une activation *softmax*, pour produire les probabilités des 5 classes.
- **Fonction de perte :** *categorical crossentropy*, utilisée pour évaluer la différence entre les probabilités prédites et les labels réels.
- **Optimiseur :** *Adam*, qui offre une convergence rapide grâce à son mécanisme d’adaptation des taux d’apprentissage.

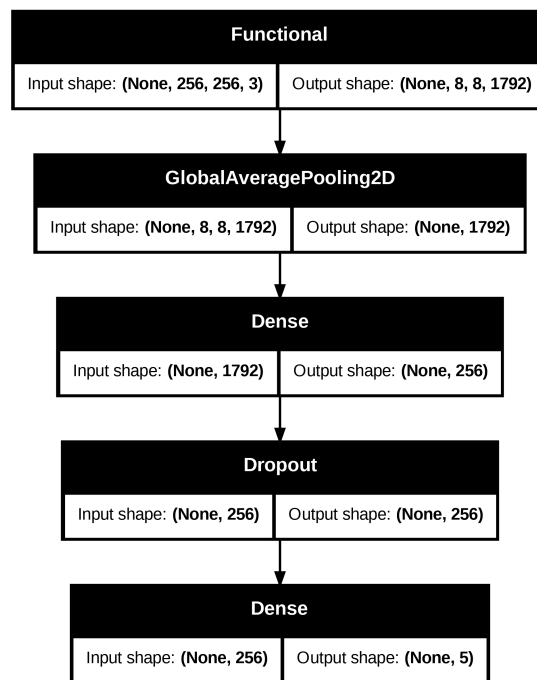


FIGURE 2 – Couches ajoutées à EfficientNetB4.

## Résultats

### Performance du modèle

Après 10 époques d’entraînement avec et sans augmentation, nous avons atteint une accuracy de validation de **94.8%**. Le tableau ci-dessous résume les résultats :

Configuration	Accuracy	Loss
Sans augmentation	91.2%	0.23
Avec augmentation (finale)	94.8%	0.18

TABLE 1 – Performances du modèle selon la configuration.

## Discussion

### Défis rencontrés

- **Problèmes de mémoire** : le nombre des images consomment beaucoup de RAM. Nous avons optimisé l'utilisation en batchant les données et en utilisant des callbacks pour libérer la mémoire après chaque époque.
- **Équilibrage des classes** : certaines classes (comme les mitochondries) étaient sous-représentées, rendant difficile leur classification.

### Limites et perspectives

- Ajouter des techniques de régularisation comme le mixup pour réduire l'overfitting.
- Étendre la méthode à une segmentation 3D complète des images FIB/SEM.

## Conclusion

Nous avons atteint notre objectif d'une accuracy de **95%** grâce à l'utilisation de techniques avancées de deep learning. Ce projet illustre la puissance des modèles préentraînés combinés à des augmentations adaptées pour des données complexes et spécialisées comme celles issues de la microscopie électronique.