

# Création de jeux de données pour l'entraînement de réseaux de neurones

Pierre MINIER

Université de Bordeaux

14 septembre 2023

# Introduction : stage en Espagne

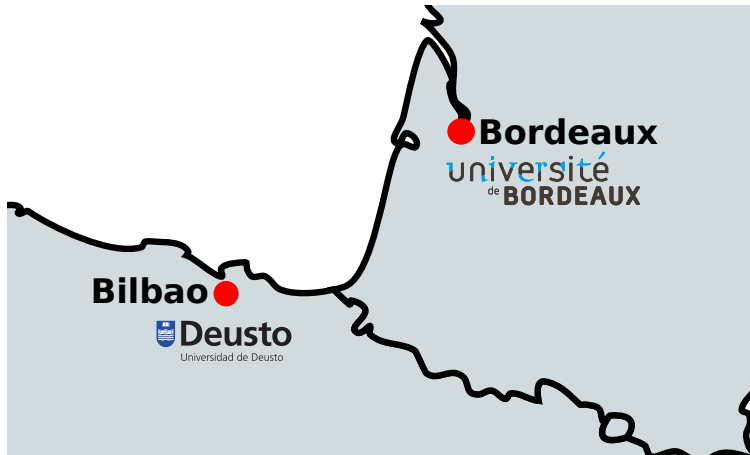
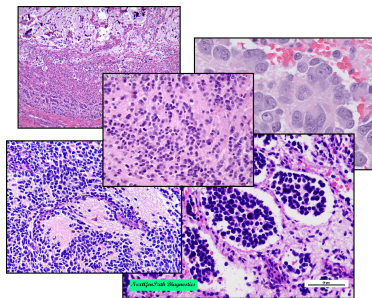
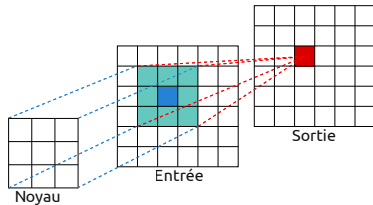


Figure 1 – Université de Deusto, Bilbao (ES)

# Introduction : objectifs du stage



(a) Pré-traitement des images



(b) Architecture adéquate de CNN

Figure 2 – Classification de tumeurs

# Sommaire

- 1 Définitions et contexte
- 2 Découpe d'images
- 3 Partition en jeux de données
- 4 Expérience

# Base de données vs. Jeux de données

## Base de données

Ensemble des images labélisées.

## Jeux de données

Partitions de la base de données.

- Entraînement ( $\sim 70\%$ ) : optimisation du modèle
- Validation ( $\sim 15\%$ ) : contrôle durant l'optimisation
- Test ( $\sim 15\%$ ) : évaluation du modèle optimisé

Nécessité d'indépendance et de diversité dans les exemples.

# Pourquoi découper les images ?

- Entrée CNN : 224x224 ou 299x299
- Certaines images : 4000x3000

# Pourquoi découper les images ?

- Entrée CNN : 224x224 ou 299x299
- Certaines images : 4000x3000

Avantages	Inconvénients
Pas de perte de détails liée à une compression	Perte de bandes de pixels latérales
Démultiplication du nombre d'éléments de la base de données	Perte de labels sur certaines découpes sans cellules observables

# Pourquoi découper les images ?

- Entrée CNN : 224x224 ou 299x299
- Certaines images : 4000x3000

Avantages	Inconvénients
Pas de perte de détails liée à une compression	Perte de bandes de pixels latérales
Démultiplication du nombre d'éléments de la base de données	Perte de labels sur certaines découpes sans cellules observables

## Perte de données

5% de pixels sont perdus avec les inconvénients cités.



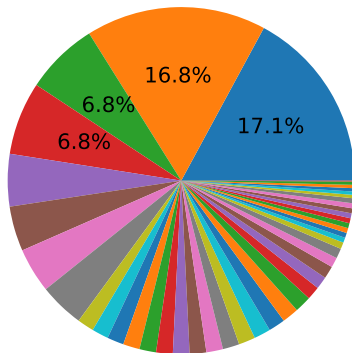
# Contraintes

## Notations

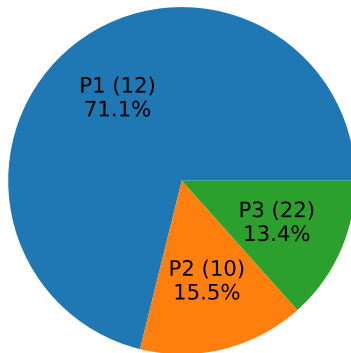
$P_i$  est la  $i$ ème partition,  $C_i$  sa capacité et  $D_i$  sa diversité.

- **Indépendance** : Les découpes d'une même image sont corrélées.
- **Capacités cibles** :  $C_1 > 0.5$  et  $C_1 + C_2 + C_3 = 1$ .
- **Diversités cibles** :  $D_1 = D_2 = D_3 = 1/3$

# Partitionnement Séquentiel (PS)



(a) Découpes viables



(b) Partition en 70-15-15

Figure 3 – Exemple d'un partitionnement séquentiel

# Partitionnement Itératif (PI)

## Algorithme (1/3)

---

### Algorithm 1 Initialisation

---

- 1:  $F$  : Liste des familles de découpes viables, triée par nombre de pixels décroissant
  - 2:  $C_j$  : Capacité restante de la partition  $P_j$
  - 3: **Tant que**  $F[1] > \min(C_2, C_3)$  **Faire**
  - 4:      $P_1 \leftarrow F[1]$
  - 5:     Mettre à jour  $C_1$
  - 6:     Supprimer  $F[1]$  de  $F$
  - 7: **Fin Tant que**
-

# Partitionnement Itératif (PI)

## Algorithme (2/3)

---

### Algorithm 2 Équilibrage des partitions

---

- 1:  $i \leftarrow 1$
  - 2: **Tant que**  $\min(|P_2|, |P_3|) < |P_1|$  **Faire**
  - 3:     **Si**  $F[i] < \min(C_2, C_3)$  **Alors**
  - 4:         Affecter  $F[i]$  à  $P_2$  ou à  $P_3$  (tour à tour si possible)
  - 5:         Mettre à jour  $C_2$  ou  $C_3$
  - 6:         Supprimer  $F[i]$  de  $F$
  - 7:     **Fin Si**
  - 8:      $i \leftarrow i + 1$
  - 9: **Fin Tant que**
-

# Partitionnement Itératif (PI)

## Algorithme (3/3)

---

### Algorithm 3 Itérations

---

- 1:  $N \leftarrow |F|$
  - 2:  $i \leftarrow 1$
  - 3: **Tant que**  $F[|F|] < \max(C_1, C_2, C_3)$  **et**  $i < N$  **Faire**
  - 4:     **Si**  $F[i] < \min(C_1, C_2, C_3)$  **Alors**
  - 5:         Affecter  $F[i]$  à  $P_1$ , à  $P_2$  ou à  $P_3$  (tour à tour si possible)
  - 6:         Mettre à jour  $C_1$ ,  $C_2$  ou  $C_3$
  - 7:         Supprimer  $F[i]$  de  $F$
  - 8:     **Fin Si**
  - 9:      $i \leftarrow i + 1$
  - 10: **Fin Tant que**
  - 11: Affecter les éléments de  $F$  à  $P_1$
-

# Partitionnement Itératif (PI)

## Paramètre d'initialisation

### Initialisation de la partition $P_1$

Soient :

- $N_1(\mu)$  : le nombre d'images initialisant la partition  $P_1$ ,
- $\mu$  : paramètre introduit, à valeur dans  $]0, 1]$ ,
- $F[i]$  : la proportion de pixels de la  $i$ ème famille dans la classe,
- $C_2$  et  $C_3$  : les capacités respectives de  $P_2$  et  $P_3$ .

$$N_1(\mu) = \left| \left\{ i \mid F[i] > \mu \times \min(C_2, C_3) \right\} \right| \quad (1)$$

# Partitionnement Itératif (PI)

Estimation du paramètre introduit (1/2)

## Fonction de perte (de coût)

Soient :

- $C_i^{eff}$  : la capacité effective pour la  $i$ ème partition
- $D_i^{eff}$  : la diversité effective pour la  $i$ ème partition

$$L(\mu) = \frac{1}{6} \left[ \sum_{i=1}^3 |C_i - C_i^{eff}(\mu)| + \sum_{i=1}^3 |D_i - D_i^{eff}(\mu)| \right] \quad (2)$$

$$\mu^* = \operatorname{argmin}_{\mu \in [0,1]} L(\mu) \quad (3)$$

# Partitionnement Itératif (PI)

Estimation du paramètre introduit (2/2)

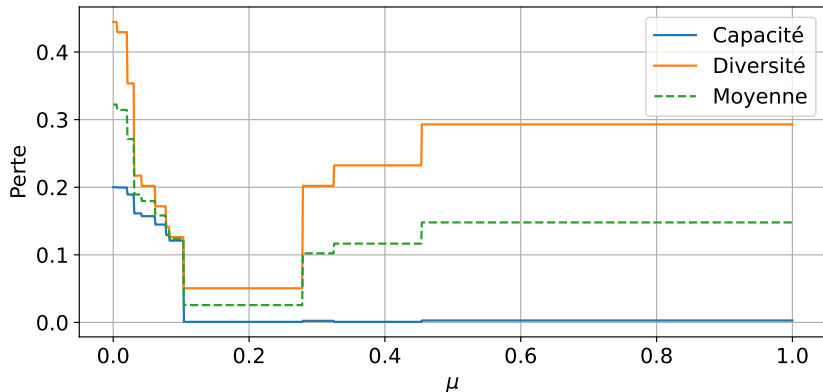


Figure 4 – Fonction de perte PI



# Partitionnement Itératif (PI)

## Résultat

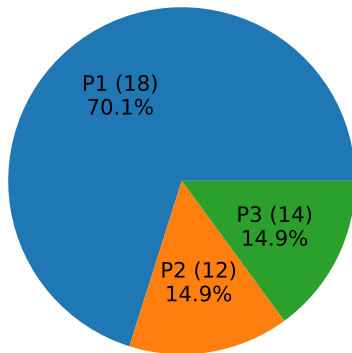
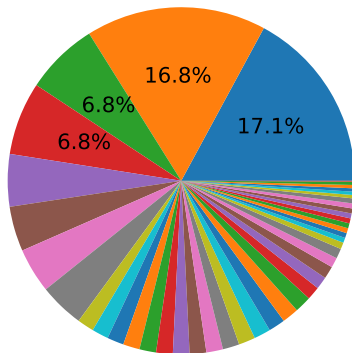
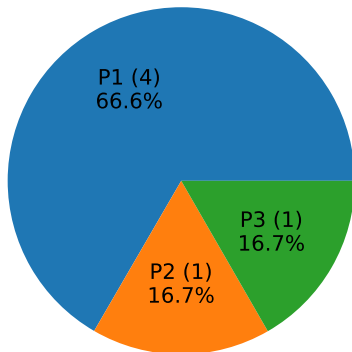


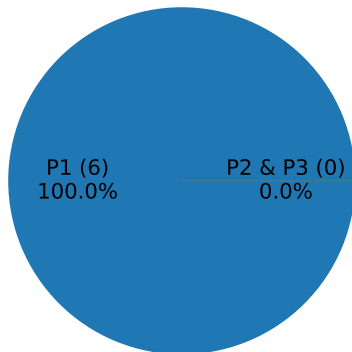
Figure 5 – Exemple d'un partitionnement itératif

# Partitionnement Itératif avec Dépassement (PID)

Cas simple mettant en échec PI



(a) Partition attendue



(b) Partition obtenue (PI)

Figure 6 – Partitionnement en 70-15-15 de 6 familles de même poids

# Partitionnement Itératif avec Dépassement (PID)

Ajustement de la fonction de perte

## Fonction de perte (de coût)

Soient :

- $C_i^{eff}$  : la capacité effective pour la  $i$ ème partition
- $D_i^{eff}$  : la diversité effective pour la  $i$ ème partition
- $\varepsilon$  : excès de capacité autorisé pour les partitions  $P_2$  et  $P_3$

$$L(\mu, \varepsilon) = \frac{1}{6} \left[ \sum_{i=1}^3 |C_i - C_i^{eff}(\mu, \varepsilon)| + \sum_{i=1}^3 |D_i - D_i^{eff}(\mu, \varepsilon)| \right] \quad (4)$$

$$(\mu^*, \varepsilon^*) = \underset{\mu \in [0,1], \varepsilon > 0}{\operatorname{argmin}} L(\mu, \varepsilon) \quad (5)$$

# Partitionnement Itératif avec Dépassement (PID)

## Optimisation

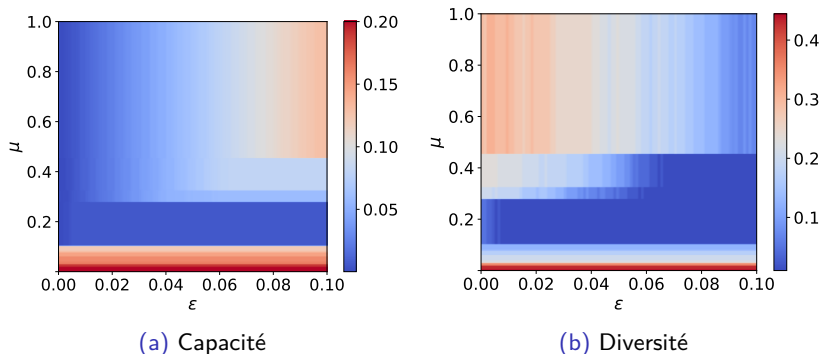
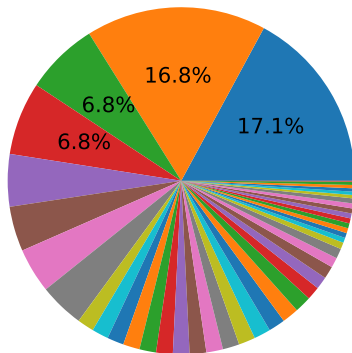


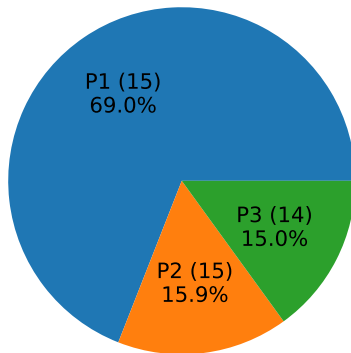
Figure 7 – Fonction de perte PID

# Partitionnement Itératif avec Dépassement (PID)

Résultat



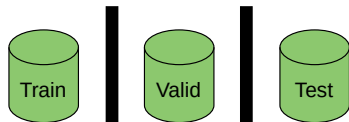
(a) Découpes viables



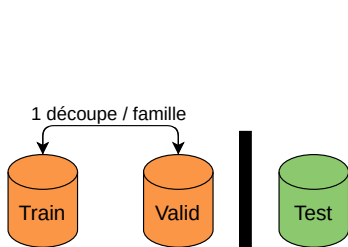
(b) Partition en 70-15-15

Figure 8 – Exemple d'un partitionnement itératif avec dépassement

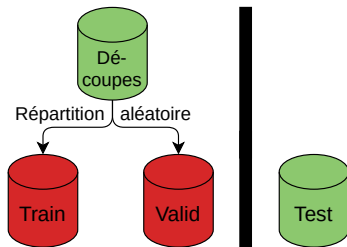
# Description



(a) Expérience 0



(b) Expérience 1



(c) Expérience 2

Figure 9 – Expériences menées

# Résultats

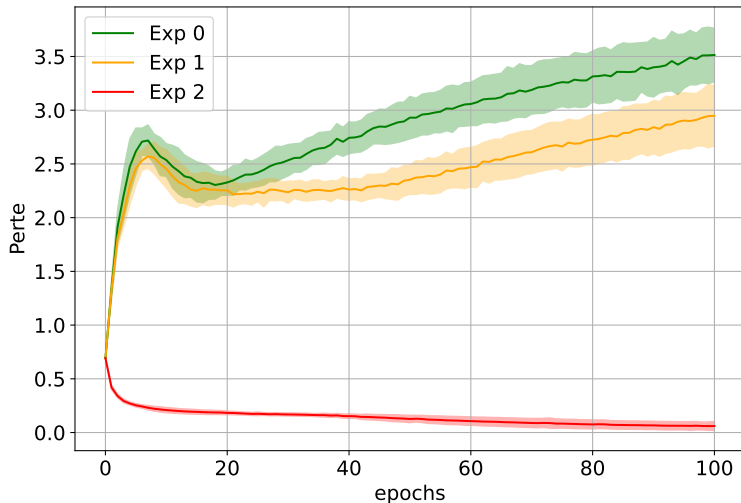


Figure 10 – Fonction de perte sur la validation pour les 3 expériences

# Conclusion

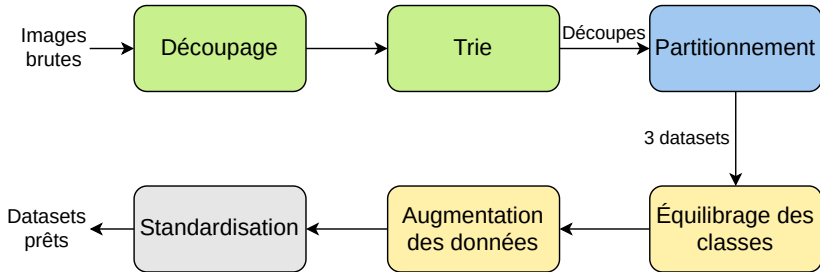


Figure 11 – Vue d'ensemble