

## **SAE Projet IA**

### **1) Flou par moyenne et flou gaussien**

Le flou par moyenne rendait une image moins nette, avec des frontières entre les différentes couleurs moins prononcées que le flou gaussien.

Nous avons donc choisi d'utiliser le flou gaussien pour le reste du projet.

### **2.1 Détection et visualisation des biomes**

1. Choix de l'algorithme de clustering le plus adapté à cette tâche. Prenez en compte notamment le nombre de pixels à traiter.

L'algorithme choisi a été celui de DBSCAN, qui, même s'il est assez lent, permet d'être assez précis et de bien gérer les formes ésotériques. Nous n'avons pas pris HAC parce que son temps d'exécution est trop élevé pour être utilisable. K-Means nécessite de connaître approximativement à l'avance le nombre de clusters idéal pour être efficace. Il n'est donc pas adapté dans le cas d'une utilisation sur des images dont on ne connaît pas le nombre de clusters.

## 2. Choix de la métrique pour calculer la similarité entre couleurs (voir TP de préparation).

Nous avons testé les 3 normes (CIELAB, Redmean, Oeil), et avons constaté que CIELAB produit le moins de clusters (écosystèmes ici), tandis que Oeil produit un résultat plus complet (image plus détaillée), mais crée beaucoup de clusters. Redmean est une solution intermédiaire, avec donc plus de clusters et une image légèrement meilleure que CIELAB, et moins de clusters et une image de moins détaillée que Oeil.

Nous utilisons les 3 normes pour chaque image, pour avoir des résultats les plus complets possibles.

## 3. Choix des paramètres du clustering. Il est conseillé de tester différentes valeurs.

Nous avons testé de nombreuses valeurs afin de déterminer l'influence de celles-ci sur le résultat, dans le but de minimiser le nombre de clusters tout en gardant une image correcte.

Ces valeurs ont été trouvées par l'expérience, après de nombreux essais.

## 4. Évaluation de la qualité du résultat obtenu. La validation peut être visuelle ou en utilisant des indices de qualité.

Les résultats obtenus étaient très satisfaisants, avec un bon découpage des différents biomes présents.

# 3 Détection et affichage des écosystèmes pour chaque biome

## 1. Choix de l'algorithme de clustering le plus adapté à cette tâche.

Notre approche a été très particulière, car nous avons effectué cette partie avant la partie 2; nous avons d'abord cherché les différents clusters (donc les différents écosystèmes ici), puis nous les avons rassemblés pour former les biomes (cf partie précédente 2.1 Détection et visualisation des biomes). Toutes ces étapes ont été effectuées avec l'algorithme DBSCAN, car le procédé que nous avons réussi à mettre en place donnait des résultats complets, avec

l'exécution d'un seul algorithme. Ce procédé avait également un temps d'exécution relativement faible, puisqu'il était aussi rapide que notre algorithme K-MEANS. La seule difficulté a été de réduire au maximum le nombre de clusters détectés, pour réduire le temps d'exécution. Cela a donc augmenté le bruit présent sur nos images.

## 2. Choix de la métrique pour calculer la similarité entre positions de pixels.

Comme dans la partie précédente, nous avons utilisé les 3 normes en parallèle pour obtenir des résultats les plus satisfaisants possible.

## 3. Choix et test des paramètres du clustering.

Comme dans la partie précédente, les différents paramètres ont été déterminés par l'expérience. Nous avons modifié les paramètres dans le but de réduire le nombre de clusters présents au maximum, tout en conservant une image acceptable, sans trop de bruit.

## 4. Évaluation de la qualité du résultat obtenu.

Nous avons réussi, en trouvant les bons paramètres, à obtenir des résultats très satisfaisants avec un temps d'exécution assez court pour un algorithme DBSCAN. En revanche, cela a parfois résulté avec des images qui avaient trop de bruit.

## Conclusion ;

Nous avons abordé le sujet de manière originale, en effectuant la détection des écosystèmes avant celle des biomes. Nous avons réussi à effectuer tout cela avec un algorithme DBSCAN, et en utilisant les 3 normes parallèlement pour comparer pour chaque image, les résultats de ces différentes normes, et donc avoir le meilleur résultat possible. Nous avons en revanche dû chercher des paramètres permettant de minimiser le nombre de clusters, et donc le temps d'exécution, qui était initialement trop élevé; cela a eu pour effet d'augmenter le bruit présent dans nos résultats. Ce procédé a permis d'obtenir des résultats satisfaisants, dans un temps relativement court.