

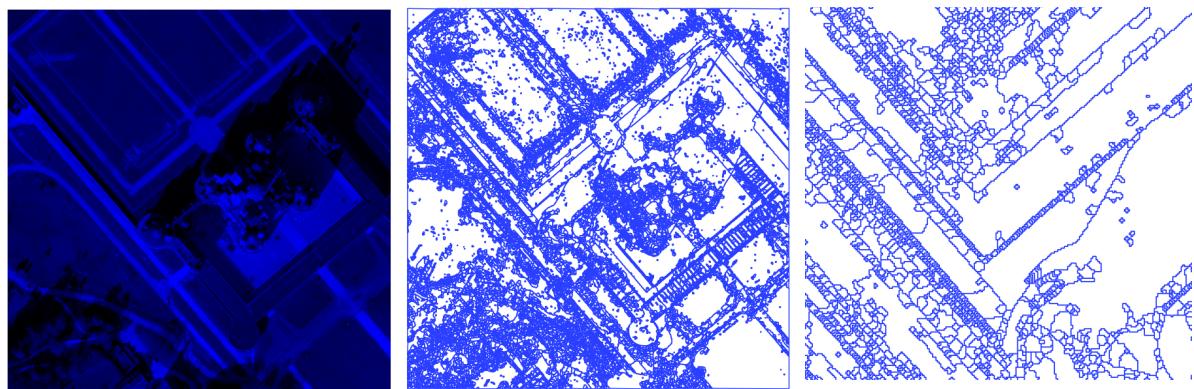
## RAPPORT DE TP : SEGMENTATION & CLASSIFICATION

A noter pour l'ensemble du rapport : les illustrations ont été mises pour avoir un rapide coup d'oeil sur les données en entrée et en sortie et ne sont généralement pas utilisés à des fins d'analyse

### **TP 1 - Segmentation**

Objectif : tester la segmentation par ligne de partage des eaux grâce à la boîte à outils Orfeo ToolBox et sa méthode watershed.

On commence par conserver un seul canal pour l'image IRC disponible du château de Chambord. Puis on peut directement exécuter l'algorithme avec les paramètres de base pour voir le premier résultat de la segmentation :



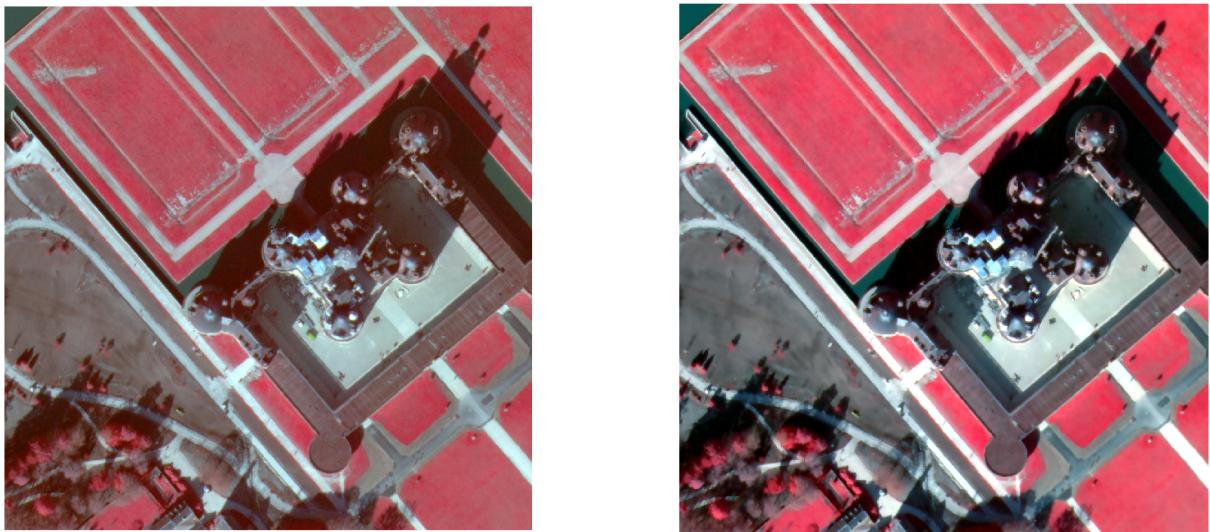
*figure 1 : image IRC canal bleu / image segmentée / zoom*

Concernant les paramètres :

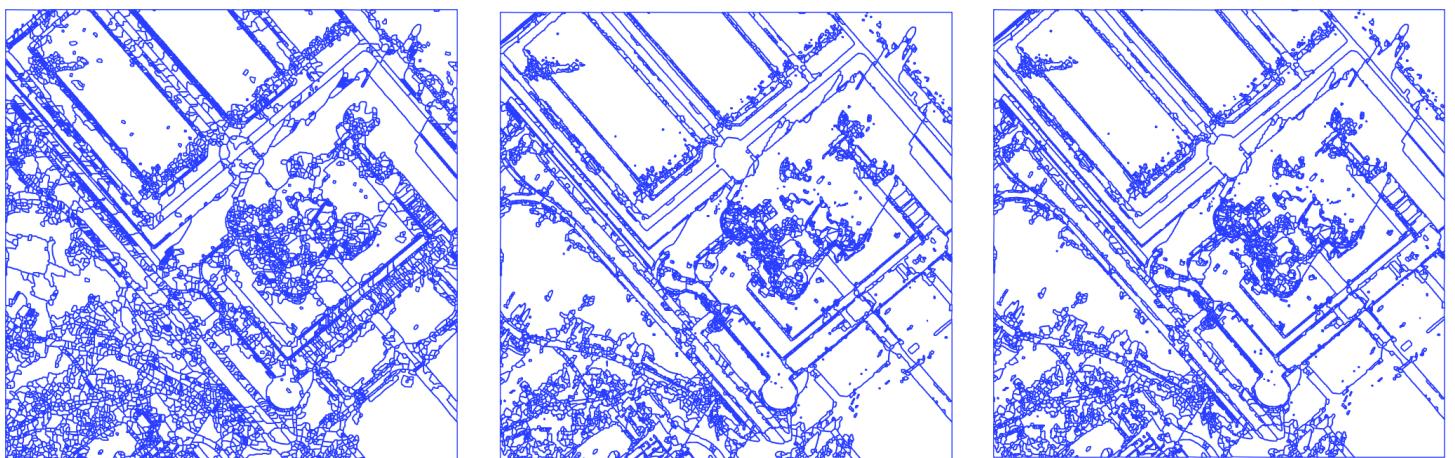
- Le Depth Threshold est concrètement le paramètre qui contrôle le niveau de sensibilité de l'algorithme aux détails fins de l'image.  
En l'augmentant, l'algorithme deviendra moins sensible aux bruits et aux détails ; ainsi on risque ici la sous-segmentation alors qu'en la diminuant de trop, c'est la sur-segmentation.
- Le Flood Level est lui un paramètre global et s'agit d'un seuil en niveau de gris à partir duquel l'algorithme commence à remplir les régions voisines à partir des marqueurs. Ainsi, augmenter ce seuil permet de remplir les régions avec des niveaux de gris plus élevés, ce qui peut potentiellement donner lieu à des régions plus grandes et plus homogènes. En revanche, le diminuer c'est potentiellement donner lieu à des régions plus petites et plus hétérogènes.

En parallèle et avant la segmentation, on pourrait appliquer des pré-traitements à l'image pour améliorer les résultats. En voici deux exemples :

- Filtre Gaussien : Le lissage de l'image avec un filtre gaussien peut aider à réduire le bruit et à améliorer les résultats de segmentation.
- Le rehaussement de contours. En effet, cela peut aider à mettre en évidence les contours des objets dans l'image, ce qui facilite donc la segmentation.



*figure 2 : image IRC / exemple image IRC après smoothing*



*figure 3 : filtre gaussien / moyenne / après rehaussement des contours*

A savoir que :

- Les illustrations placées dans les figures servent pour visualiser rapidement les différences de résultats de segmentation avec ces paramètres et donc mettent en avant l'efficacité du rehaussement de contours par rapport à celle du filtre gaussien uniquement pour ces paramètres.
- D'ailleurs, ces paramètres sont volontairement laissés comme tels dans les algorithmes car l'objectif était seulement d'observer les améliorations possibles qu'un pré-traitement aurait sur l'efficacité de la segmentation pour cette image du château de Chambord..

On ne propose donc pas de solution optimale ici.

## TP 2 - Analyse OBIA

### Exercice 2.1 : Régularisation d'une classification pixellaire avec une segmentation

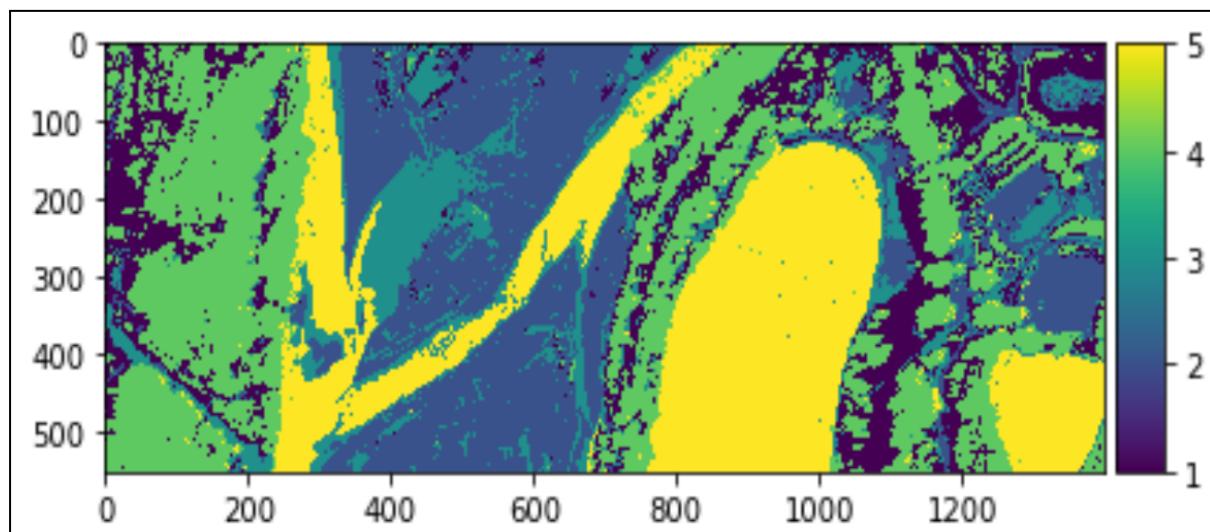
Objectif : chaque pixel de l'image de classification est réaffecté à la classification majoritaire dans le segment auquel il appartient.

Pour cela, on aura alors besoin en entrée d'une image de classification et une de segmentation, de plus la sortie espérée sera une nouvelle image de classification.

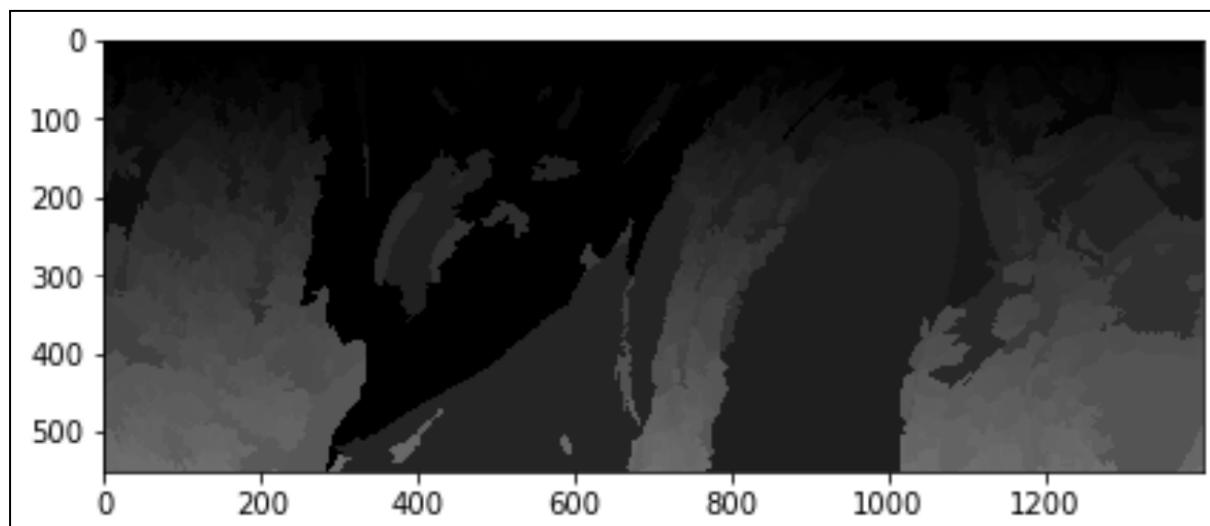
Concernant le comment :

- étape 1 : on récupère les classes des pixels appartenant au même segment :  
En parcourant chaque pixel et donc en récupérant la classe et le segment auquel appartient ce pixel, on récupère ainsi les index de cette classe et ce segment pour finalement incrémenter de 1 à une matrice M à la base vide de dimension (nombre de classes, nombre de segments) à l'endroit que désigne les index. Ainsi, on compte pour chaque valeur de classe leur population au sein de chaque segment.
- étape 2 : on trouve quelle est la classe majoritaire pour chaque segment :  
En parcourant chaque segment, on récupère l'index de la classe majoritaire en cherchant la valeur maximale au sein de chaque colonne de la matrice M ; puis comme précédemment, on crée un vecteur V de taille (nombre de segments) pour stocker ces index. Ici, on obtient alors la classe majoritaire pour chaque segment.
- étape 3 : on crée l'image de sortie de dimension identique à l'image de classification d'entrée et finalement attribué à chaque pixel, la classe majoritaire dans le segment auquel il appartient :  
En parcourant chaque pixel, on retrouve le segment et son index auquel il appartient puis grâce à ce dernier, on fait cette attribution en retrouvant la classe majoritaire dans le vecteur V qui stockait cette information. Ainsi, on obtient cette nouvelle image de classification pixellaire qui a été régularisée avec une image de segmentation.

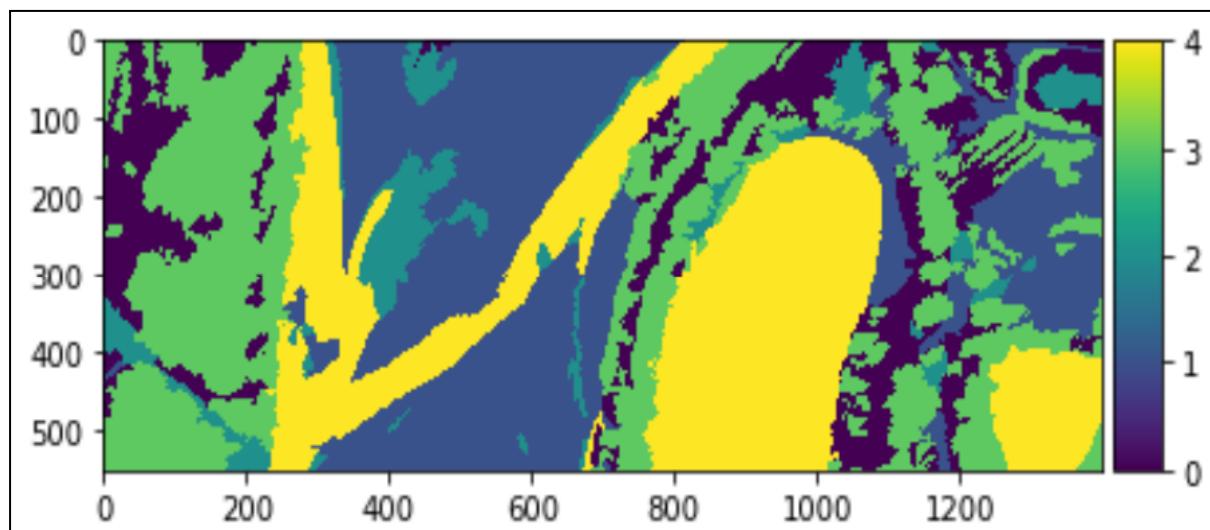
*Observation du résultat :*



*figure 1 : image de classification en entrée*



*figure 2 : image de segmentation en entrée*



*figure 3 : nouvelle image de classification en sortie*

Exercice 2.2 : Régularisation des attributs pour une classification

Objectif : chaque pixel de l'image de segmentation doit être réaffecté à la valeur moyenne de sa radiométrie dans le segment auquel il appartient.

Pour cela, on aura alors besoin en entrée d'une image IRC et une de segmentation, de plus la sortie espérée sera une nouvelle image IRC, cette fois-ci moyennée.

Concernant le comment :

- étape 1 : en parcourant les segments, on retrouve tous les pixels qui appartiennent à chaque segment
- étape 2 : on calcule ensuite les moyennes radiométriques pour ce segment, et donc sur ce groupe de pixels obtenu à l'étape précédente
- étape 3 : on attribue finalement à tous les pixels dans le segment la valeur moyenne de sa radiométrie

*Observation du résultat :*

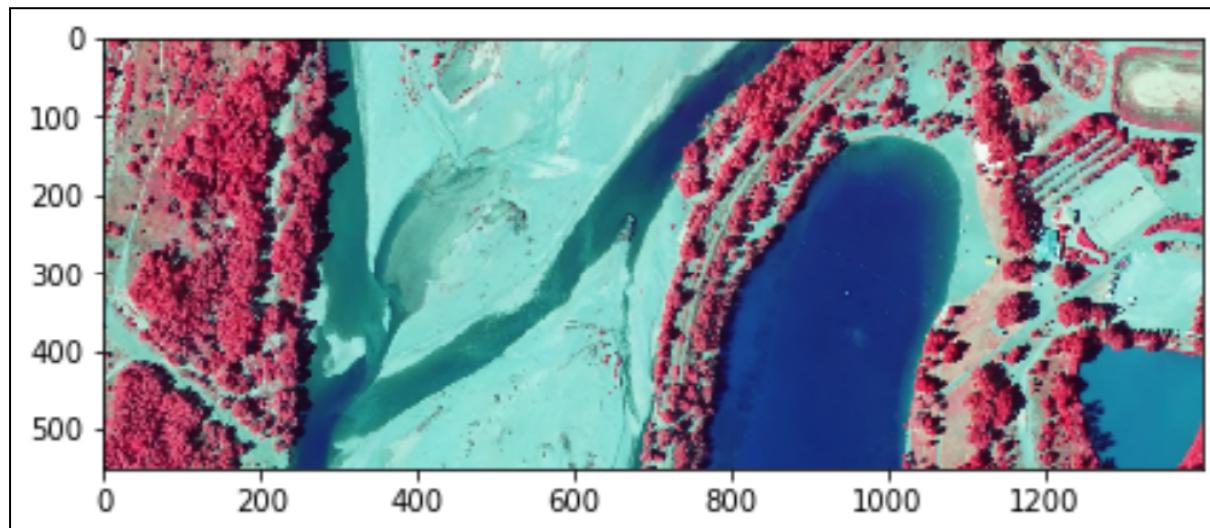


figure 1 : image IRC en entrée

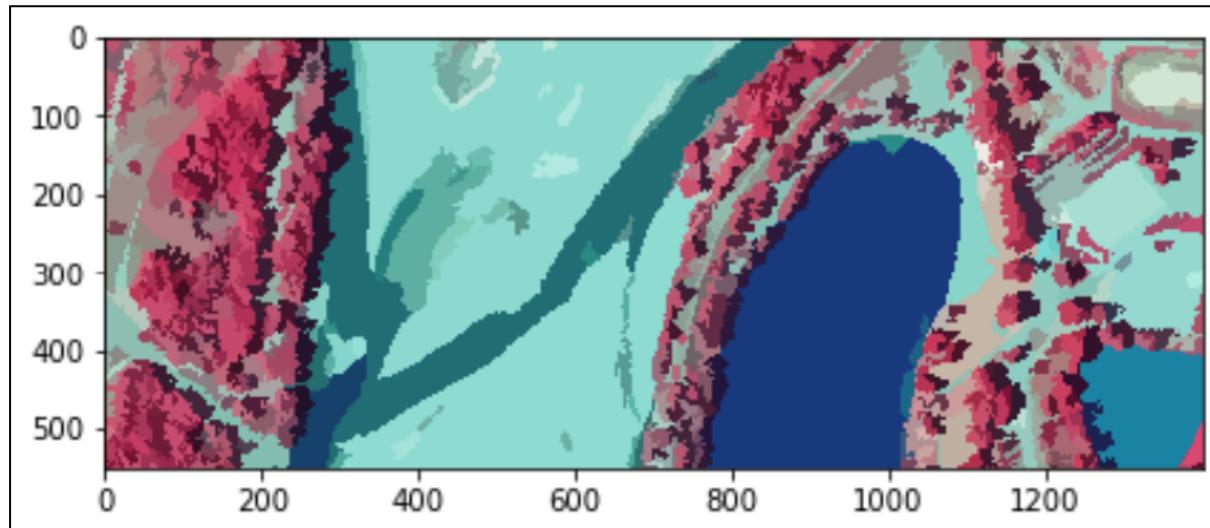
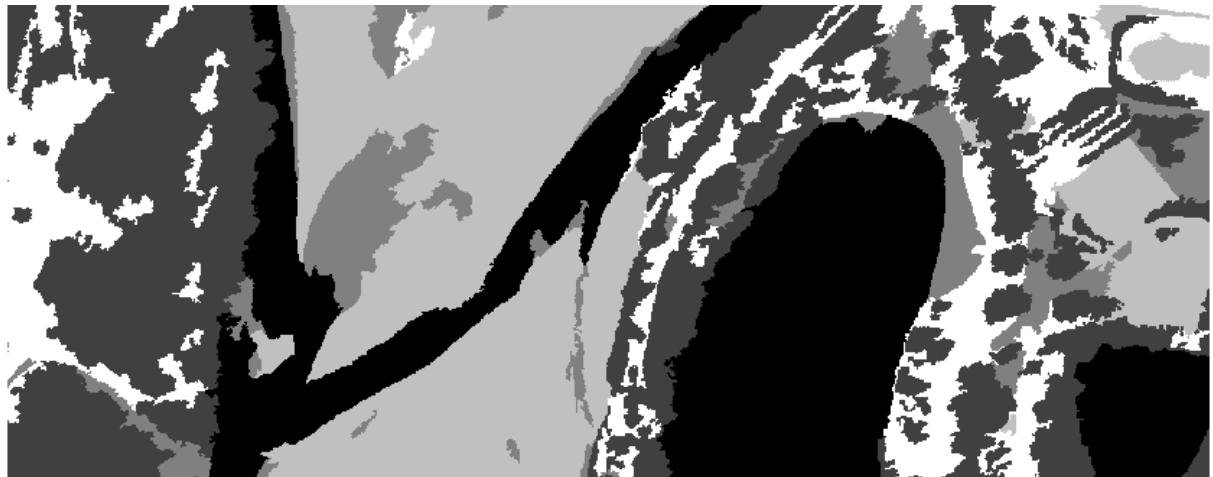


figure 2 : image IRC moyennée en sortie

Deuxième étape : avec l'image IRC moyennée et un modèle de classification donné, on utilise l'outil ImageClassifier pour directement obtenir notre image classée.

On aura alors un autre résultat de classification en suivant un autre processus, ce qui nous pourra nous mener à les comparer.



*figure 3 : classification à l'aide de ImageClassifier et du modèle donné*

On peut alors déjà observer que la classification que l'on vient de réaliser semble effectivement plus lisse, plus homogène sur les différentes grandes parties de l'image IRC. Ce qui semble bien logique vu que l'on obtient cette classification à partir d'une image IRC moyennée, qui semble aussi plus lisse et plus homogène.

Pour les comparer mathématiquement, on réalise une carte des différences binaire grâce à la calculatrice raster. Le résultat binaire met alors en évidence directement les différences de classification entre celle-ci en figure 3 et celle en sortie de l'exercice 1 de ce même TP.



*figure 4 : carte des différences binaire entre cette classification et le résultat du 2.1*

On pourrait aussi réaliser une carte des différences classe par classe en donnant une formule spéciale à la calculatrice raster.

Dans la figure 5, on a utilisé :  $10 \times \text{classif\_exo2} + \text{classif\_exo1}$

Ainsi, en sachant que les classes de la première classification vont de 0 à 4 et que celles de la deuxième vont de 1 à 5 puis en allant voir dans les fréquences de l'histogramme de cette nouvelle classification, on observe :

- Pour les valeurs 10, 21, 32, 43 et 54, ce sont les classes similaires entre les deux classifications
- Pour les autres, ce sont les différences de classification ; par exemple, on observe que les différences les plus importantes se trouvent pour la troisième classe. Les presque mille occurrences pour la valeur 31 veulent ainsi dire que 1000 pixels classés dans la troisième pour la deuxième classification sont classés dans la deuxième pour la première classification.

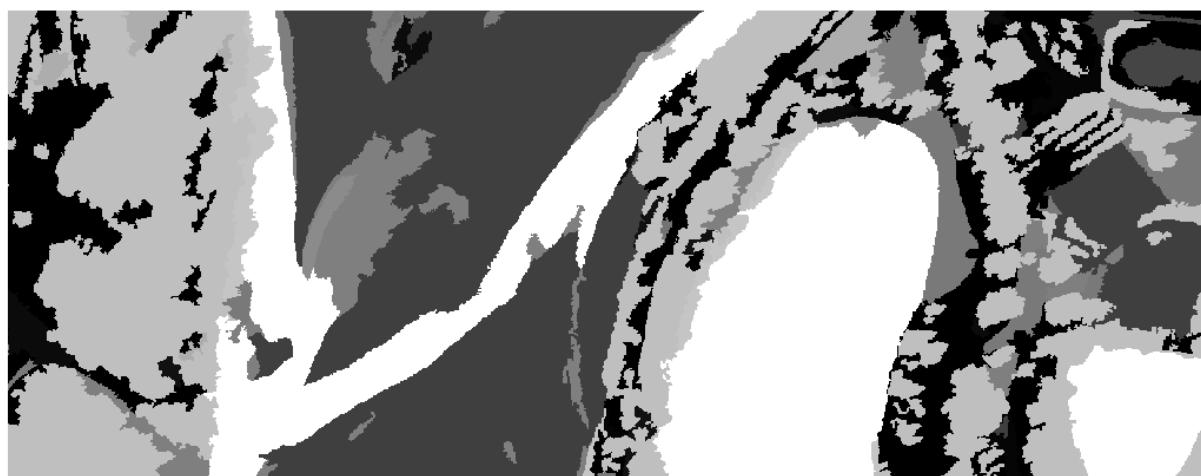


figure 5 : carte des différences classe par classe

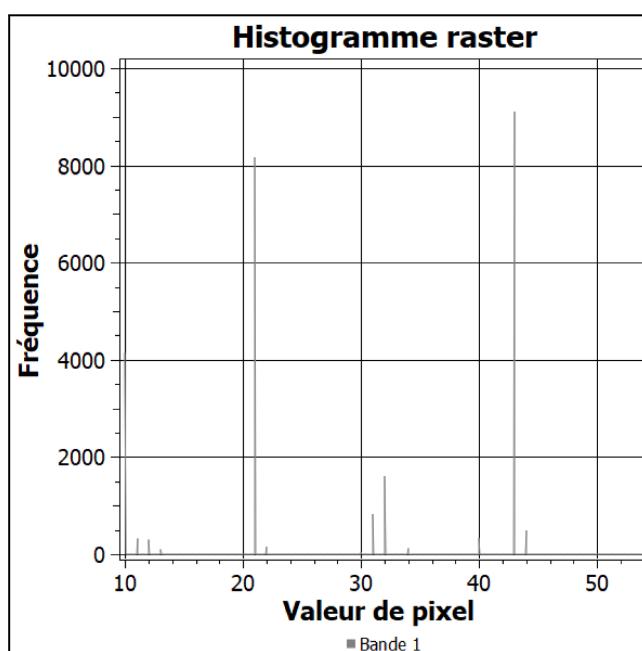


figure 6 : histogramme de la carte des différences classe par classe

Exercice 2.3 : Amélioration sémantique d'une classification

Objectif : tous les pixels d'un segment sont affectés à une classe qui dépend de la proportion des classes présentes dans le segment.

Pour cela, on a en entrée une image de classification et une de segmentation, de plus la sortie espérée est une nouvelle image de classification régularisée.

Malheureusement pour cette partie, qui est ici plus libre vu que l'on a l'opportunité de choisir nous même nos critères pour la classification, rien d'incroyable n'a été tenté.

J'ai bien pensé et repensé à des moyens de gérer la proportion dans un segment et d'appliquer ensuite plus à la main, de nouvelles classes ou quand la classe est vraiment écrasante de majorité alors on garde cette classe pour le segment.

Finalement, ce qui compte le plus, ce sont les résultats mais rien n'était vraiment probant puisque les résultats ressemblaient vraiment à la classification en sortie dans la figure 1 qui a elle été trouvée avec la classe majoritaire au sein de chaque segment.

Je conçois alors que le manque de résultat implique le même résultat que si je n'avais pas cherché, mais pour le coup, il serait vraiment intéressant de voir quelles méthodes ont été trouvées, réalisées et qui démontrent de vrais résultats.

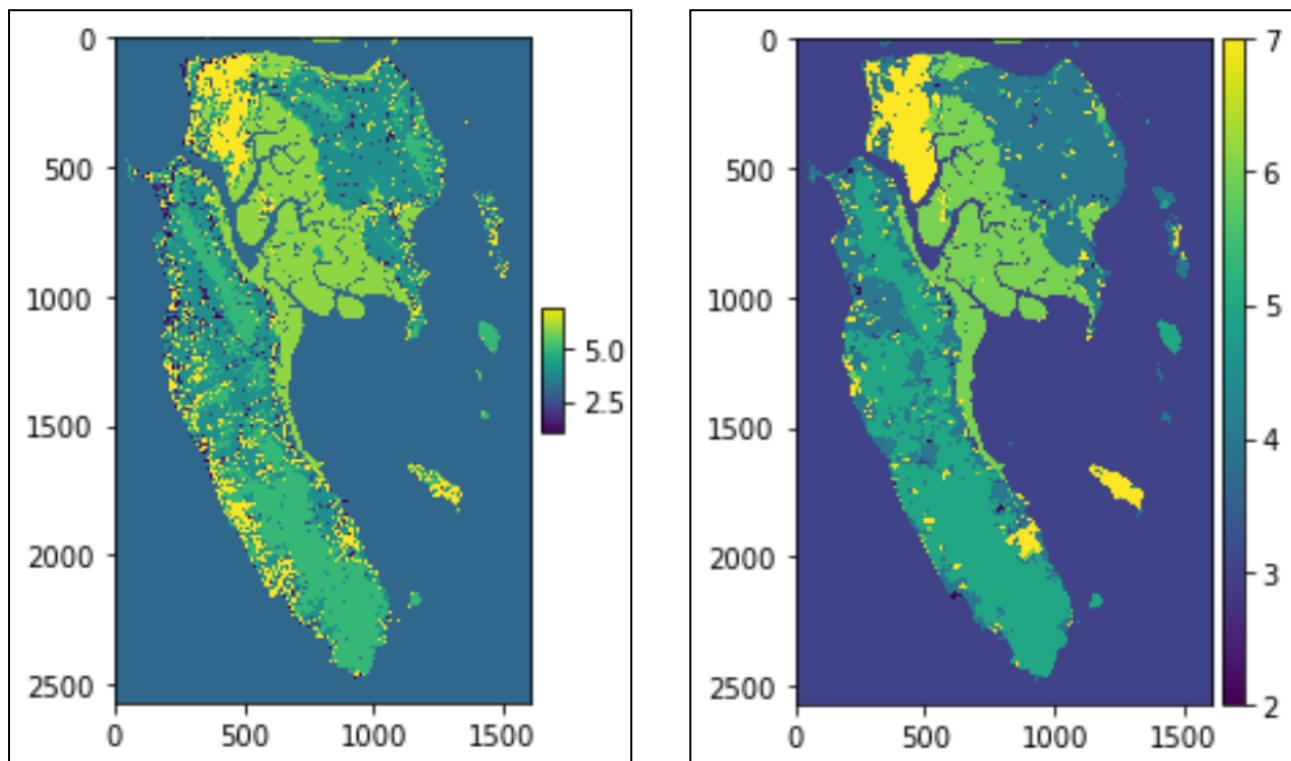


figure 1 : image de classification en entrée / image de classification première sortie