

TP : Fuzzy C-means

Nicolas Pluven and Gabriel Lucchini

Lien Github : https://github.com/Leirbagg/reco_forme_TP1

1. Introduction

La segmentation d'image est une étape essentielle en vision par ordinateur et en traitement d'image. Elle permet de diviser une image en régions distinctes, permettant la reconnaissance d'objet par exemple. Parmi les méthodes de segmentation, il existe les approches floues, approches permettant de gérer des données imprécises, incomplètes et/ou incertaines. Nous verrons la méthode Fuzzy C-means (FCM), qui permet de classer les pixels d'une image en groupes en tenant compte de la possibilité que les pixels appartiennent à plusieurs clusters simultanément, avec différents degrés d'appartenance. Dans ce TP, l'objectif est d'implémenter l'algorithme Fuzzy C-Means en Python et de l'appliquer à la segmentation d'images en régions floues. Plus précisément, il s'agit de réaliser une segmentation à n clusters pour des images en niveaux de gris et en couleur, en produisant des cartes de chaleur (heatmaps) pour visualiser les résultats. Ce rapport présente les différentes étapes de l'implémentation, les résultats obtenus, ainsi qu'une discussion sur les avantages et les limites de l'approche.

2. Algorithme Fuzzy C-Means

L'algorithme Fuzzy C-Means est une méthode de classification floue qui attribue à chaque point de données un degré d'appartenance à plusieurs clusters. Il fonctionne selon les étapes suivantes :

1 - Initialisation des clusters : Des centres de clusters sont initialisés aléatoirement.

2 - Calcul des distances et mise à jour de la matrice d'appartenance : La distance entre chaque point et chaque centroïde est calculée et pour chaque point, un degré d'appartenance à chaque cluster est calculé en fonction de sa distance aux centroïdes. Les degrés d'appartenance sont mis à jour à chaque itération.

3 - Mise à jour des centroïdes : Les nouveaux centroïdes sont calculés en pondérant les points par leurs degrés d'appartenance.

4 - Critère d'arrêt : L'algorithme itère jusqu'à ce que les centroïdes ne changent plus de manière significative (selon un seuil de tolérance : $1e-5$) ou qu'un

nombre maximum d'itérations soit atteint (100 par défaut).

3. Implémentation

3.1. Choix du langage de programmation

Pour l'implémentation de cet algorithme nous avons choisi python en raison de ses bibliothèques pour le traitement d'images tel que *skimage* et *matplotlib*. La bibliothèque *numpy* nous aide aussi pour l'implémentation et les opérations numériques plus précisément.

3.2. Description du code

L'algorithme Fuzzy C-Means pour la segmentation d'image a été implémenté dans la fonction *fuzzy_c_means* selon les étapes décrites précédemment dans la partie 2 avec une étape en plus pour l'appliquer aux images. On commence par identifier si l'image est en niveaux de gris ou en couleur. Puis on passe au FCM :

- Initialisation des centroids : on initialise les centroids en sélectionnant des pixels aléatoires dans l'image.

- Calcul des distances : on calcule la distance euclidienne entre chaque pixels et centroids. (Pour les images en niveaux de gris, une distance absolue est utilisée, tandis que pour les images RGB, la distance euclidienne entre les vecteurs de couleur est calculée)

- Mise à jour de la matrice d'appartenance : on calcule la matrice d'appartenance en fonction des distances. Chaque pixels a un degrés d'appartenance à tout les clusters. Pour éviter les divisions par zéro, une petite constante est ajoutée.

- Mise à jour des centroïdes : on calcule les nouveaux centroids en pondérant les pixels par leurs degrés d'appartenance. Cette étape est répétée jusqu'à ce que les centroïdes convergent ou qu'un nombre maximal d'itérations soit atteint.

- Application à une image : la fonction *load_image* permet de charger des images en niveaux de gris ou en couleurs. On applique FCM à ces images puis on visualise les résultats sous forme de heatmaps avec la fonction *plot_heatmaps*. (On a

aussi une fonction pour afficher les images segmentées à chaque étapes du FCM.

4. Résultats

4.1. Segmentation pour les images en niveaux de gris

Voici l'image en niveau de gris à laquelle on va appliquer notre algorithme FCM.



Figure 1. grayscale image

Pour notre segmentation d'image en niveau de gris avec FCM, on choisi 3 clusters. Les résultats montrent des cartes de chaleur des degrés d'appartenance des pixels à chaque cluster, illustrant les régions floues de l'image.

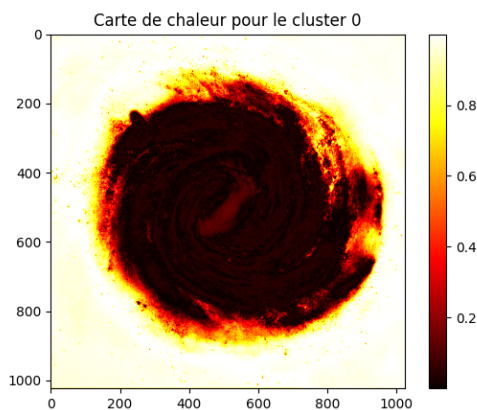


Figure 2. FCM for grayscale image, $k=3$

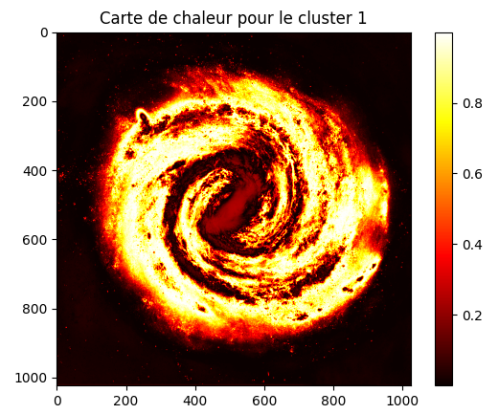


Figure 3. FCM for grayscale image, $k=3$

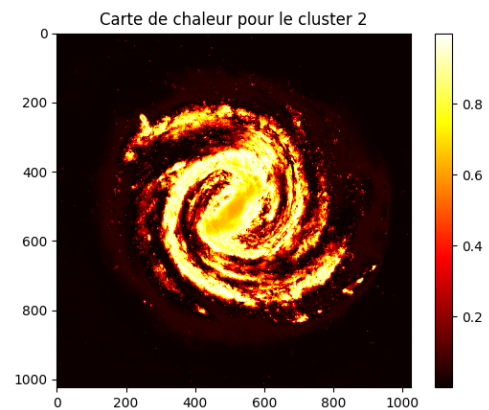


Figure 4. FCM for grayscale image, $k=3$

On obtient des résultats convaincants, avec 3 clusters bien distincts et des régions floues.

4.2. Segmentation pour les images RGB

Voici l'image RGB à laquelle on va appliquer notre algorithme FCM.



Figure 5. RGB image

L'algorithme a également été appliqué à des images en couleur, où chaque pixel est considéré comme un vecteur dans l'espace RGB. On choisit aussi 3 clusters. Le résultat montre une segmentation plus détaillée et complexe, avec des clusters représentant des variations de couleurs dans l'image. Les résultats montrent des cartes de chaleur des degrés d'appartenance des pixels à chaque cluster, illustrant les régions floues de l'image.

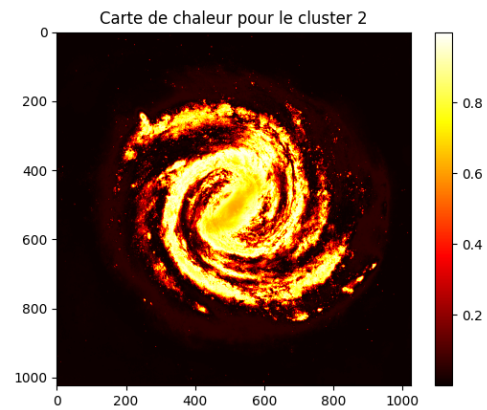


Figure 8. FCM for RGB image, $k=3$

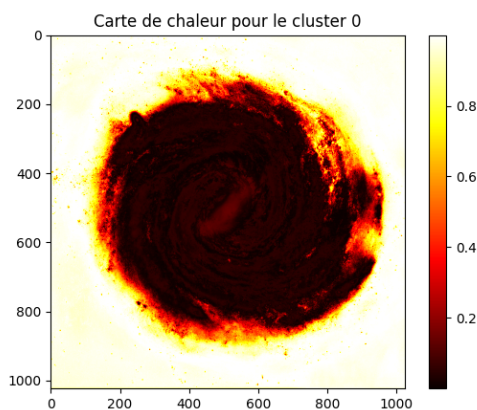


Figure 6. FCM for RGB image, $k=3$

5. Discussion et Analyse

Les résultats montrent que l'algorithme FCM est bien adapté aux images présentant des transitions progressives entre les régions. Les cartes de chaleur et les images segmentées illustrent la capacité de FCM à gérer les régions floues. Aussi, on remarque que nos clusters en fonction du FCM pour images en niveaux de gris ou en couleur sont très similaires. En modifiant le paramètre m , on observe que plus il est proche de 1, plus l'algorithme FCM va faire un traitement similaire à k -means. De plus, de la même manière que k -means, le choix du nombre de clusters reste essentiel.

6. Conclusion

Ce travail pratique a permis de mettre en œuvre et d'appliquer l'algorithme Fuzzy C-Means à la segmentation d'images en niveaux de gris et en couleur. Les résultats montrent que FCM est une méthode puissante pour capturer les régions floues des images, en attribuant un degré d'appartenance multiple aux pixels.

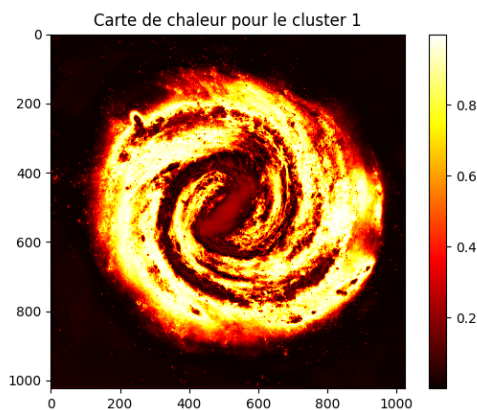


Figure 7. FCM for RGB image, $k=3$