



Rapport TP : Fuzzy C-means

Pour la segmentation d'images

Réalisé par :
Rebai Mohamed Younes

[Le Repo Git du TP](#)

[Le Google Colab du TP](#)

Date : 25 octobre 2024

1 Introduction

Le Fuzzy C-means (FCM) est un algorithme de classification floue utilisé dans la segmentation d'images. Contrairement aux méthodes traditionnelles de classification comme K-means, où chaque point de données appartient à un seul cluster, FCM permet à chaque point d'avoir une appartenance floue à plusieurs clusters simultanément. Cela rend cet algorithme particulièrement utile dans les cas où les frontières entre objets ne sont pas clairement définies, comme dans la segmentation d'images à faible contraste ou avec des transitions graduelles.

2 Principe de l'algorithme Fuzzy C-means

L'algorithme suit plusieurs étapes clés pour réaliser la segmentation d'une image :

- **Initialisation** : L'algorithme commence par initialiser aléatoirement une matrice d'appartenance où chaque pixel a un degré d'appartenance à chaque cluster. La somme des appartences d'un pixel à tous les clusters est égale à 1.
- **Mise à jour des centroïdes** : À chaque itération, les centroïdes des clusters sont recalculés en prenant en compte les appartances floues des pixels. Cela permet d'ajuster progressivement les positions des clusters.
- **Mise à jour des appartences** : Une fois les centroïdes mis à jour, les degrés d'appartenance des pixels sont réévalués en fonction de leur distance aux nouveaux centroïdes. Les pixels proches d'un centroïde auront une appartenance plus forte à ce cluster.
- **Convergence** : L'algorithme s'arrête lorsque la variation des appartences d'une itération à l'autre devient suffisamment petite (en dessous d'un seuil fixé, souvent $\epsilon = 0.01$). Cela peut se produire avant d'atteindre le nombre maximal d'itérations, comme observé lorsque la barre de progression s'arrête avant 100%.



FIGURE 1 – Convergence avant le nombre maximal d'itérations

L'algorithme se poursuit jusqu'à la convergence ou jusqu'à ce que le nombre maximal d'itérations soit atteint.

3 Expérimentations

L'algorithme Fuzzy C-means a été appliqué à différentes images de la Voie lactée, en niveaux de gris et en couleur (RGB), avec différents nombres de clusters. Les résultats sont présentés sous forme de heatmaps (cartes de chaleur) qui montrent le degré d'appartenance des pixels à chaque cluster, ainsi que des images segmentées résultantes.

3.1 Gray scale Milky Way (2 clusters)

Dans cette première expérience, une image en niveaux de gris de la Voie lactée a été utilisée et divisée en deux clusters. L'objectif est de séparer l'image en deux grandes régions distinctes, correspondant par exemple aux étoiles et au fond plus sombre.

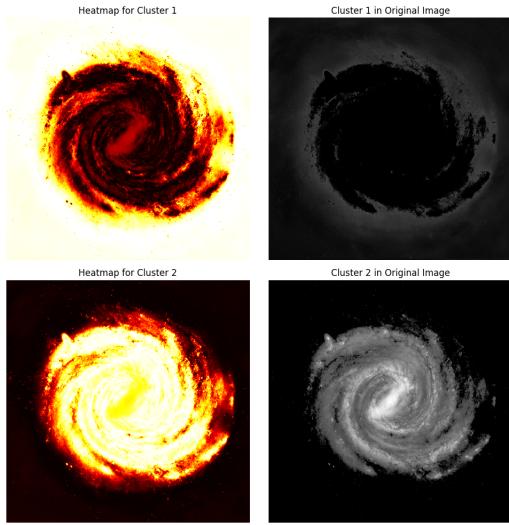


FIGURE 2 – Heatmap des appartenances pour chaque cluster (2 clusters, image en niveaux de gris)

Les heatmaps montrent les zones où l'appartenance à un cluster est dominante. Sur cette base, une image segmentée est produite.

La segmentation en deux clusters sépare grossièrement les étoiles et les zones plus sombres du fond.

3.2 Gray scale Milky Way (5 clusters)

Pour une segmentation plus détaillée, l'image en niveaux de gris a été divisée en 5 clusters. Cela permet de capturer plus de variations subtiles dans les intensités de lumière dans l'image, en séparant des régions avec des caractéristiques lumineuses différentes.

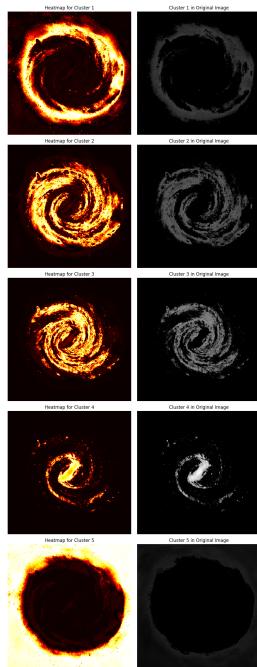


FIGURE 3 – Heatmap des appartenances pour chaque cluster (5 clusters, image en niveaux de gris)

La heatmap montre que l'image est désormais divisée en 5 zones, chacune représentant un niveau d'intensité lumineuse distinct.

L'augmentation du nombre de clusters permet de mieux capturer la complexité de la scène, avec une meilleure distinction entre les différentes zones de luminosité dans l'image.

3.3 RGB Milky Way (2 clusters)

L'expérience suivante applique l'algorithme à une image RGB (en couleur) avec seulement 2 clusters. L'objectif est de voir comment l'algorithme segmente une image en couleur en seulement deux groupes distincts.

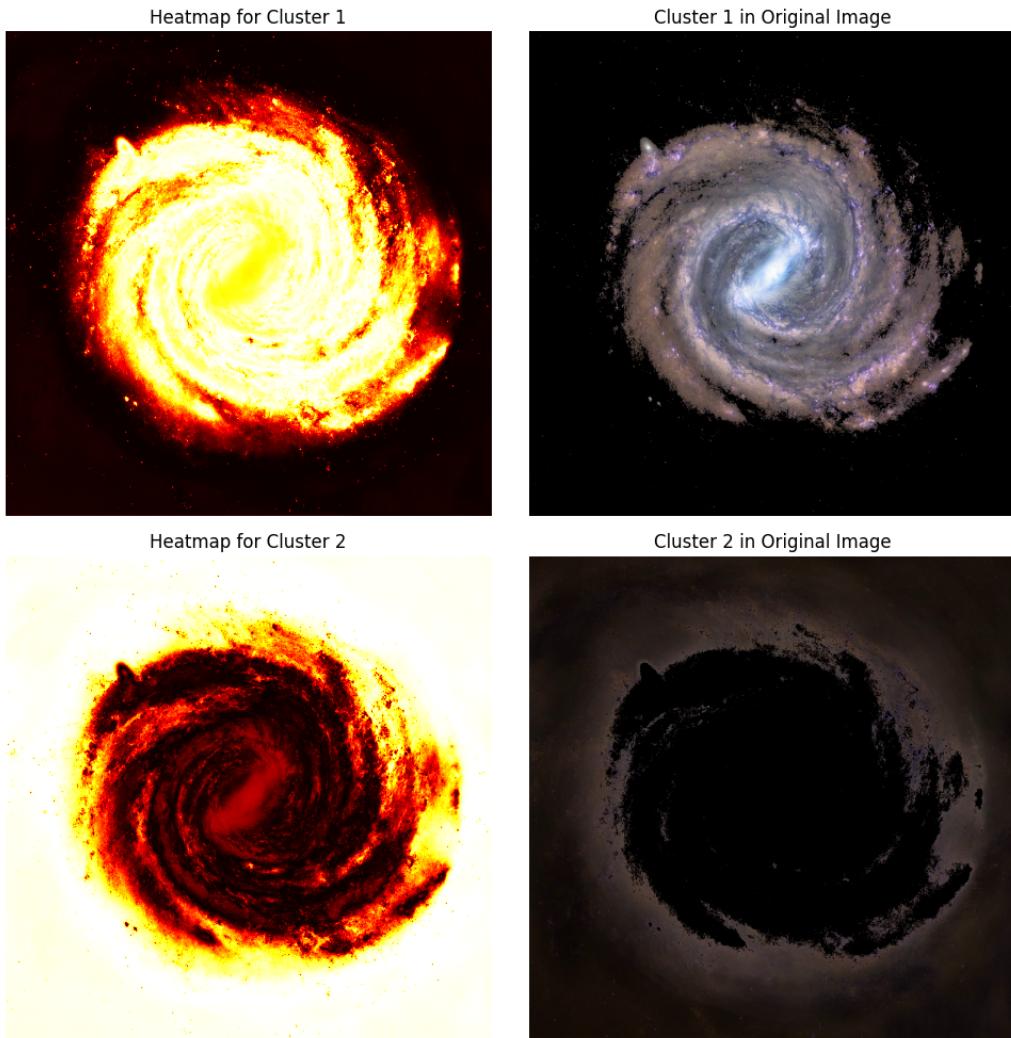


FIGURE 4 – Heatmap des appartenances pour chaque cluster (2 clusters, image RGB)

Les heatmaps montrent les régions de l'image où un cluster est dominant par rapport à l'autre.

L'image segmentée montre une séparation binaire entre les parties les plus brillantes et les plus sombres de l'image. Cela peut être utile pour détecter de larges structures dans des images complexes.

3.4 RGB Milky Way (5 clusters)

Enfin, l'algorithme est appliqué à la même image en couleur, mais cette fois avec 5 clusters. Cela permet de capturer une plus grande diversité de couleurs et de niveaux de luminosité dans l'image.

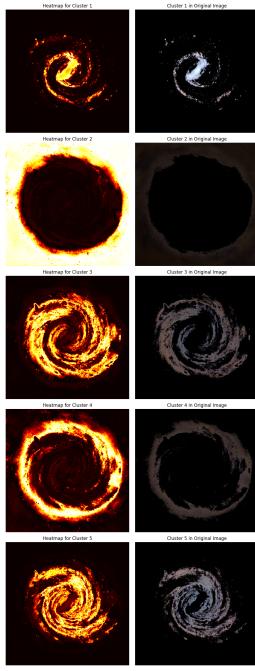


FIGURE 5 – Heatmap des appartences pour chaque cluster (5 clusters, image RGB)

Les heatmaps des 5 clusters montrent les différentes zones de l'image avec des caractéristiques de couleur ou de luminosité distinctes.

L'image segmentée montre plus de détails, avec une meilleure différenciation des structures lumineuses et des nuances dans l'image.

3.5 Image du chat (3 clusters)

Enfin, l'algorithme a été appliqué à une image d'un chat avec 3 clusters. Cette segmentation permet de diviser l'image en trois groupes distincts, correspondant aux différentes régions de couleur et de texture.

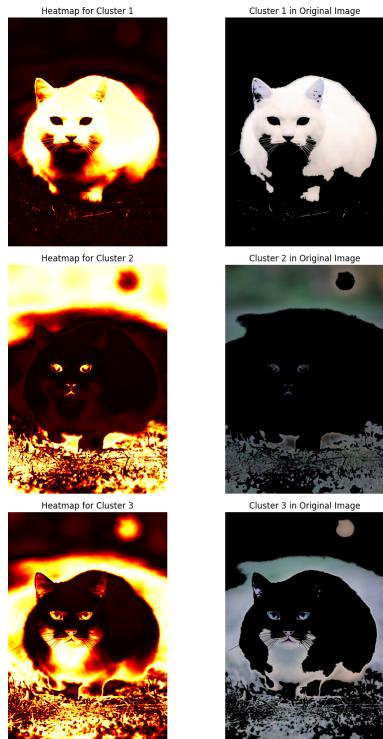


FIGURE 6 – Heatmap des appartences (3 clusters, image du chat)

4 Segmentation Colorisée des Clusters

Une amélioration supplémentaire appliquée aux résultats du clustering flou est la segmentation colorisée des clusters. Dans cette méthode, chaque région de cluster de l'image se voit attribuer une couleur représentative calculée à partir de la couleur moyenne des pixels de cette région. En assignant des couleurs uniques à chaque cluster, on crée une segmentation visuellement distincte et codée par couleur de l'image, permettant d'identifier clairement les différentes régions en fonction de leurs associations de cluster.

Un seuil est appliqué à cette carte de chaleur pour déterminer le cluster dominant de chaque pixel. La couleur moyenne des pixels de chaque cluster est ensuite appliquée à la région correspondante du cluster, aboutissant à une image segmentée simplifiée et informative qui facilite l'interprétation visuelle du clustering.

Cette approche est particulièrement utile dans des applications telles que la détection d'objets, l'imagerie médicale et la manipulation photo, où les zones segmentées fournissent des informations claires sur la structure de l'image et les régions d'intérêt.

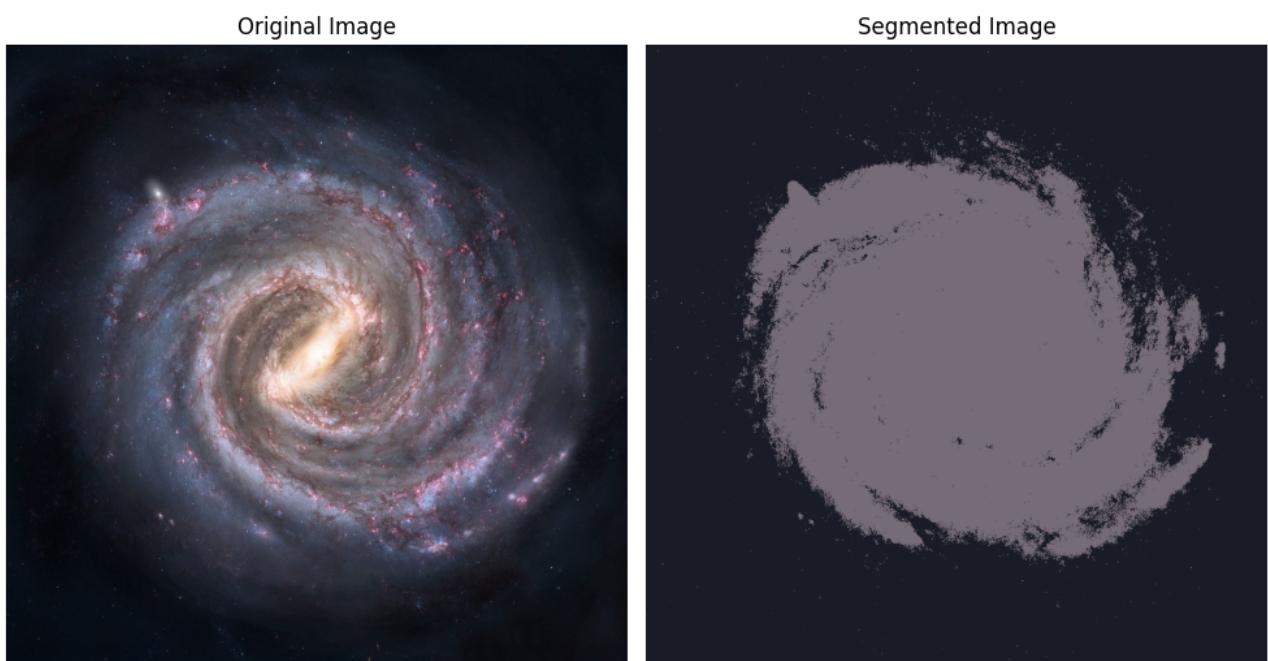


FIGURE 7 – Résultat (2 clusters, image RGB)

5 Conclusion

Le Fuzzy C-means s'est avéré être une méthode efficace pour la segmentation d'images, en particulier dans les cas où les frontières entre les objets sont floues ou difficiles à distinguer. L'augmentation du nombre de clusters permet de capturer davantage de détails dans l'image, au prix d'une complexité accrue. De plus, l'algorithme peut converger avant d'atteindre le nombre maximal d'itérations, comme le montre la barre de progression qui s'arrête parfois avant 100%, indiquant que la solution a déjà convergé.