Telecom-Paris

TRAVAUX PRATIQUES - IMA203

TP4

Modèles déformables

Réalisé par:

SAIFEDDINE BARKIA

FIRAS DAKHLI

TAHER ROMDHANE

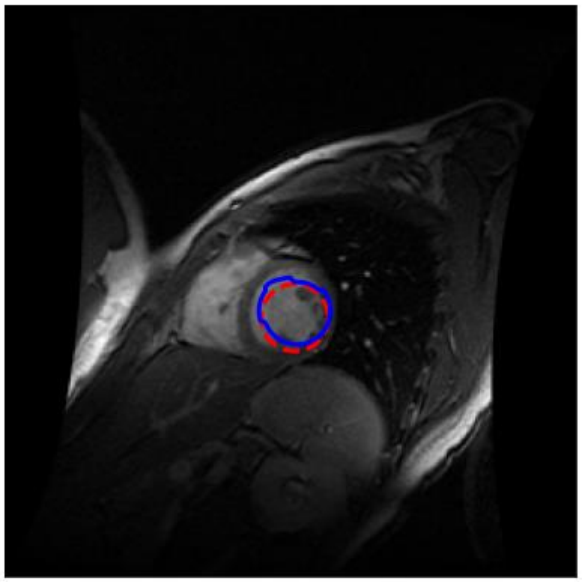
1.Paramètres

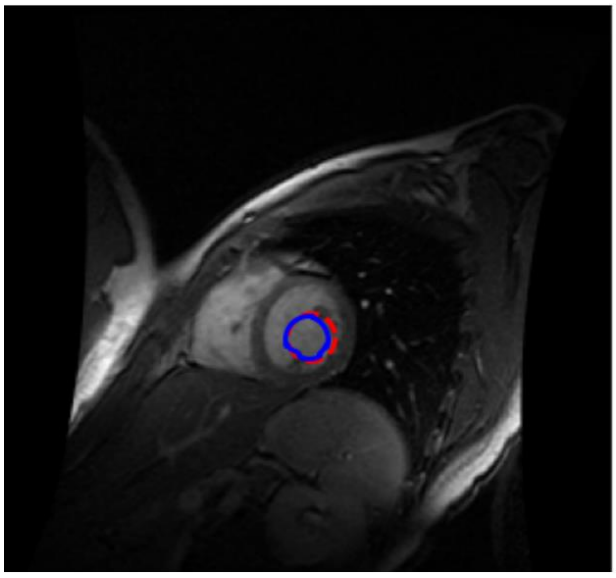
1.1 Méthodes de contours actifs paramétriques

Dans cette partie, nous allons essayer de deviner l'influence de chaque paramètre en les prenant aux alentours des conditions aux limites**.**

- **Init :**

L'objet qu'on souhaite segmenter à une forme circulaire, il est donc raisonnable de choisir un cercle comme un contour fermé initiale.





```
s = np.linspace(0, 2*np.pi, 100)
r = 140 + 15*np.sin(s)
c = 130 + 15*np.cos(s)
init = np.array([r, c]).T
```

```
s = np.linspace(0, 2*np.pi, 100)
r = 140 + 10*np.sin(s)
c = 130 + 10*np.cos(s)
init = np.array([r, c]).T
```

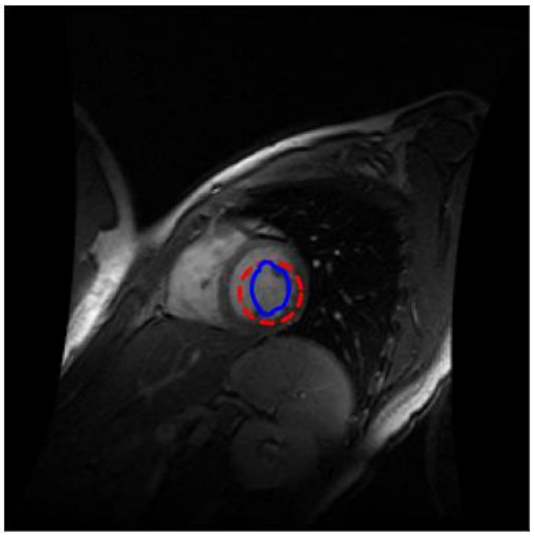
Tout d'abord, nous avons constaté que pour l'algorithme de Snakes à contours actif , la convergence de la région voulu dépend fortement de l'initialisation de des contours pris. On remarque que si on change l'initialisation, la solution change complétement et on obtient une segmentation totalement différente. (Tout en gardant tous les autres paramètres fixes.)

Dans le deuxième exemple, on remarque qu'on ne converge pas vers la région qu'on souhaite obtenir si on prend le contour initiale totalement à l'intérieur de la région qu'on souhaite détecter.

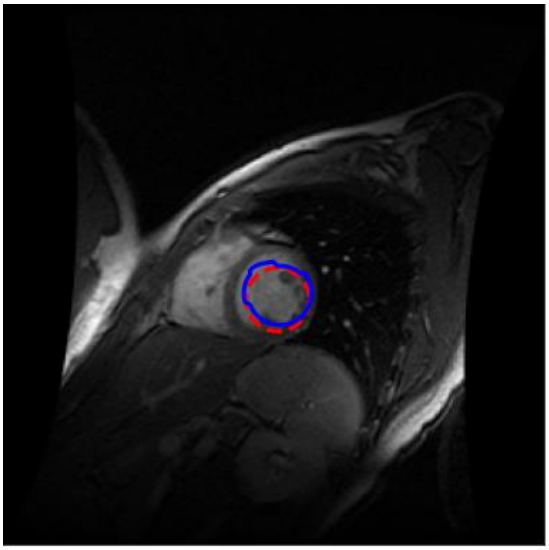
Remarque : comme vu en cours, on peut surmonter ce problème en ajoutant, soit une force de ballon , soit la technique de gradient diffusé pour que l'algorithme soit moins dépendant de l'initialisation prise.(ce ci peut être aussi fixer en jouant avec le paramétre de w_line ou w_edge qui définissent l'énergie de l'image).

- **Alpha :**

Lorsque nous avons varié les paramètres alpha, on remarque qu'il s'agit d'un paramètre qui indique la longueur du serpent. En effet, des valeurs plus élevées accélèrent la contraction du serpent.



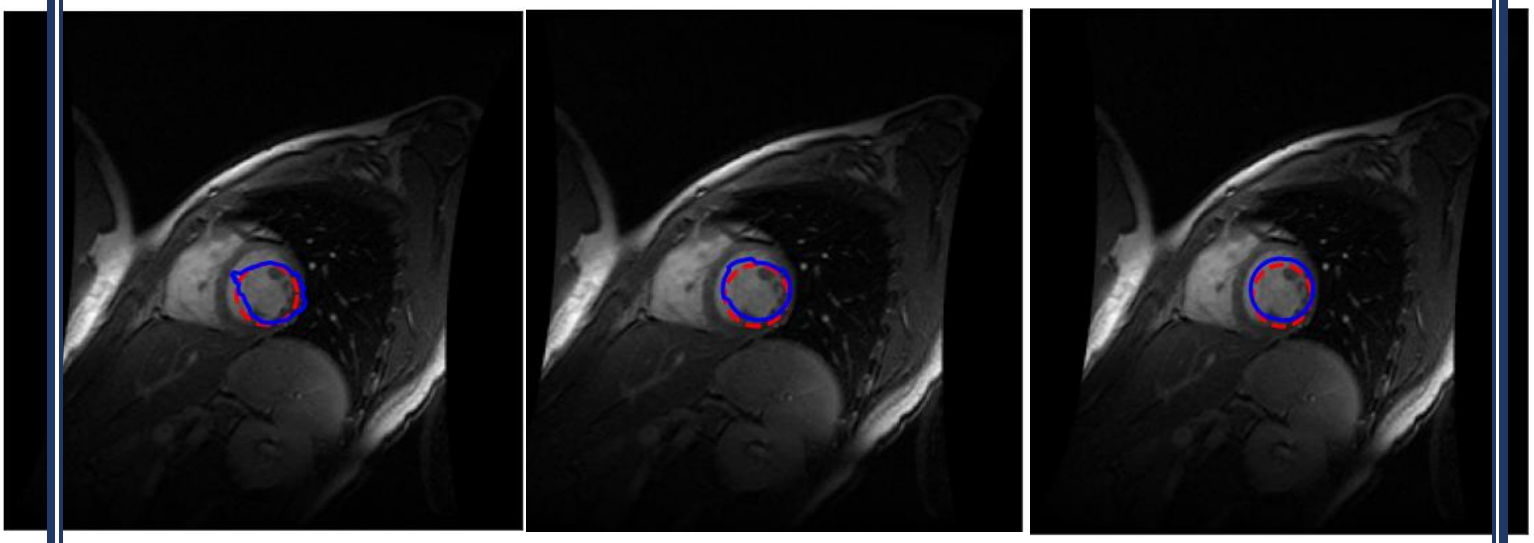
Résultat pour alpha =100



Résultat pour alpha = 0.5

Or lpha est élevée, le serpent e

- **Beta :**
  Paramètre de forme de « lisseté » (souplesse) du serpent.



-

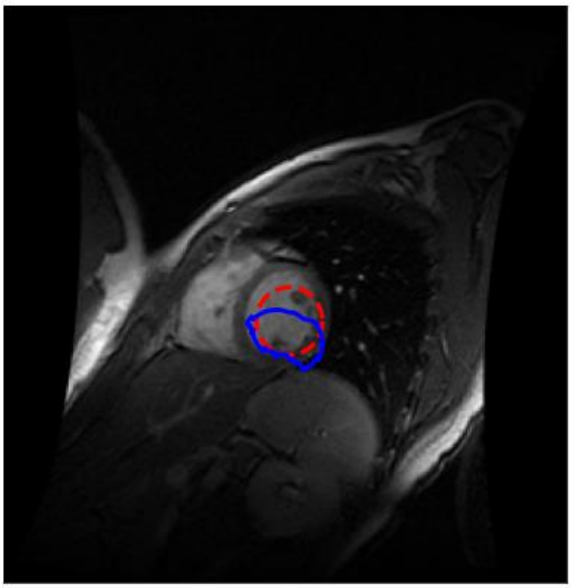
Résultat pour beta = 0 Résultat pour beta = 5 Résultat pour beta = 100

On remarque qu'on prenant une béta = 0, le résultat finale à tendance à développer des coins (points pointus). On remarque bien que plus beta augmente, plus le serpent devient plus lisse, autrement dit, on se rapproche plus de la forme d'un cercle)

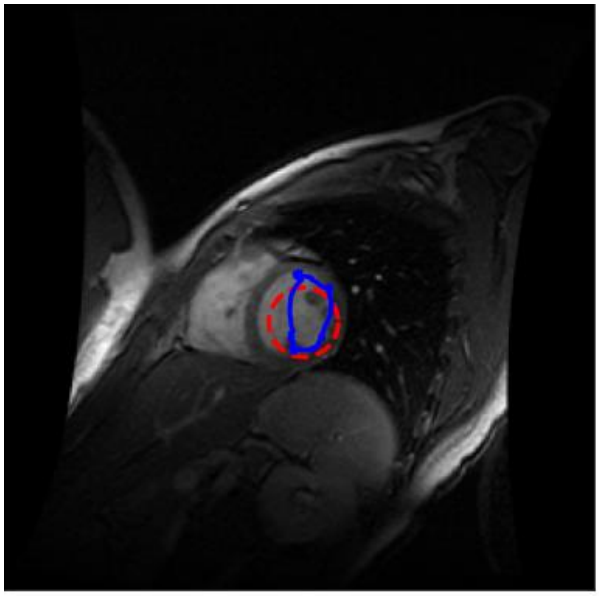
- **W_line :**

Contrôle l'attraction pour la luminosité. Utilisez des valeurs négatives pour attirer vers les régions sombres.

Selon le signe de W_line, le serpent sera attiré soit par les lignes claires, soit par les lignes sombres. Sous réserve de ses autres contraintes, le serpent essaiera de s'aligner sur le contour voisin le plus clair ou le plus sombre.



Résultat pour - W_line = -100

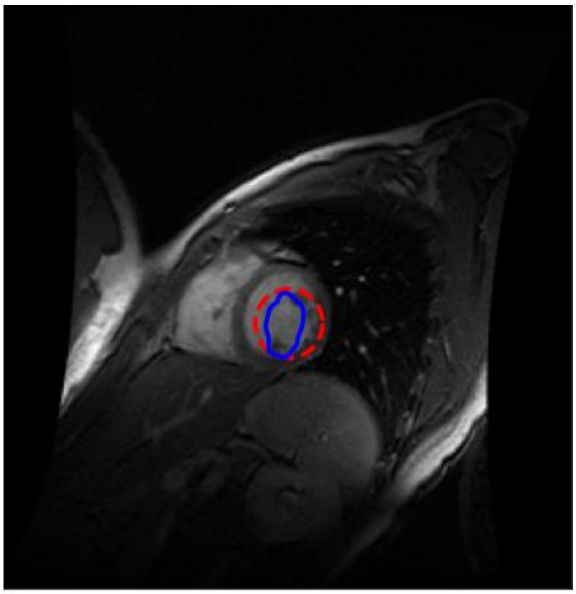


Résultat pour - W_line = 100

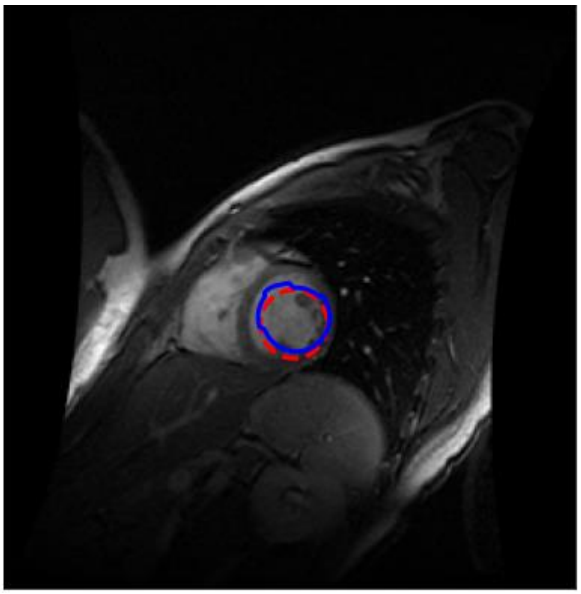
Pour W_line = -100 , on remarque que les contours sont attiré par les lignes sombres et pour W_line = 100 , les contours sont attirés vers une ligne +- claires.

- **W_edge :**

Contrôle l'attraction vers les bords. Si on utilise des valeurs négatives, on repousse le serpent des bords.Si on utilisent des valeurs positives ,le« snake » est attiré par les contours ayant des gradients élevé.



Résultat pour W_edge = -2

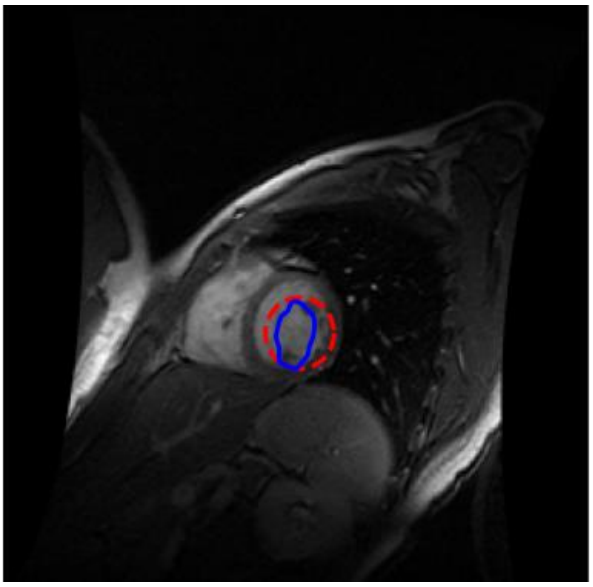


Résultat pour W_edge = 20

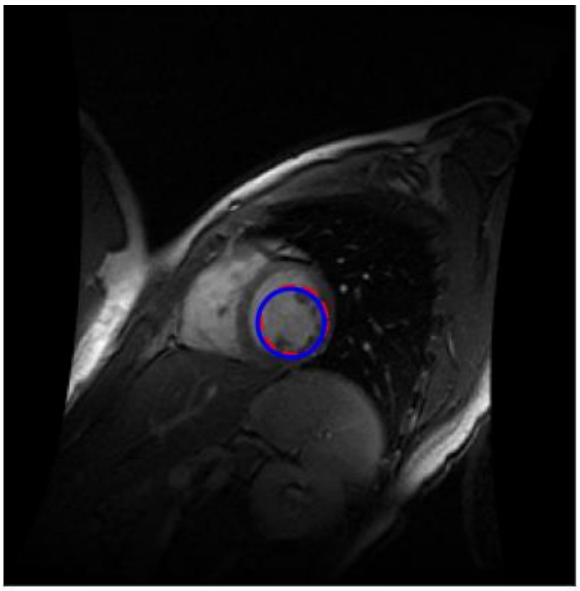
On remarque que lorsque W_edge est négative, on diverge de la zone qu'on veut initialement segmenter. Donc on s'éloigne du bord de cette région.

- **Gamma :**

Paramètre de pas de temps explicite. Si on met gamma=0, cela désactive tous les effets du gradient de l'image.



Résultat pour W_edge = -2, gamma = 0.001



Résultat pour W_edge = -2, gamma =0

En prenant gamma = 0, on désactive l'effet de gradient. En effet , même si on a pris W_edge négative, on remaruqe qu'on n'a pas éloigner des bords de la région à segmenter.

1.2 Méthode de Chan et Vese

Modèle de contour actif en faisant évoluer un jeu de niveaux. Peut être utilisé pour segmenter des objets sans limites clairement définies.

- **Mu :**

Paramètre de «longueur de bord». Des valeurs de mu plus élevées produiront un bord «rond», tandis que des valeurs plus proches de zéro détecteront des objets plus petits.

Résultat pour Mu = 0.25

Résultat pour Mu = 0.9

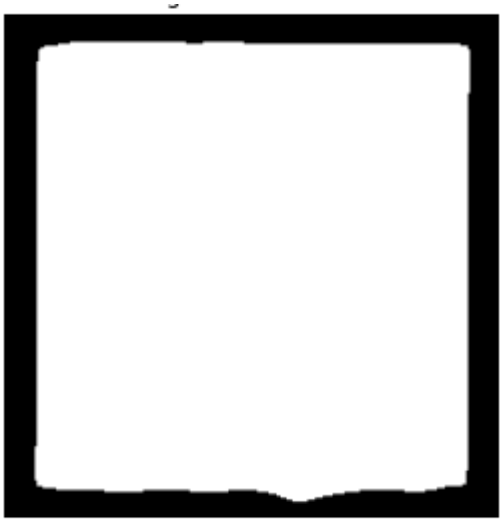
On remarque que lorsque mu est petite, on détecte beaucoup plus d'objet de l'image que lorsque mu est grande.

- **Lambda :**

Les paramètres sont reliés entre eux.

-**Lambda 1** : est le paramètre de différence «par rapport à la moyenne» pour la région de sortie avec la valeur «True». Si elle est inférieure à lambda2, cette région aura une plus grande plage de valeurs que l'autre.
-**Lambda 2** : est le paramètre de différence «par rapport à la moyenne» pour la région de sortie avec la valeur «False». Si elle est inférieure à lambda1, cette région aura une plus grande plage de valeurs que l'autre.

Résultat pour lambda1 = 5, lambda2 = 1

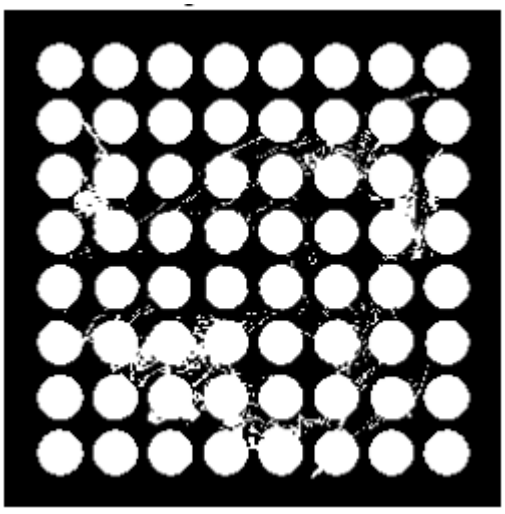


Résultat pour lambda1 = 4, lambda2 = 8

On remarque bien que lorsque lambda 2 < lambda1 , la région dans laquelle les valeurs sont « false » (=0 :c'est-à-dire noires) est plus grande que celle dans laquelle les valeurs sont « true » (=1 : c'est-à-dire blanches). Même pour des valeurs de lambda très élevé, on obtient, une image qui n'est pas raisonnable , il faut faire attention en utilisant ces deux paramètres pour la méthode de Chan et Vese

- **TOL :** Tolérance de variation de niveau définie entre les itérations. Si la différence de norme L2 entre les ensembles de niveaux d'itérations successives normalisées par la zone de l'image est inférieure à cette valeur, l'algorithme supposera que la solution a été atteinte.

Résultat pour Tol = 0.001



Résultat pour Tol = 0.05

C’est une condition d’arrêt, on remarque que pour une valeur élève de tol, l’algorithme donne n’importe quoi comme résultat , il converge dans une seule itération.

- **Max_iter**

C’est le nombre maximal d'itérations autorisé avant que l'algorithme ne s'interrompe. Donc il est évident que plus max_iter est élevée, plus on obtient un résultat de segmentation mailler mais par contre on perd au niveau de complexité de l’algorithme.

- **Dt :**

C'est un facteur de multiplication qui sert à accélérer l'algorithme pour les calculs de chaque étape. Bien que des valeurs plus élevées puissent accélérer l'algorithme, des problèmes de convergence peuvent également être induits.

Résultat pour Dt = 1000

Résultat pour Dt = 0.5

On remarque que l’exécution de l’algorithme est plus vite , mais les régions segmenter ne sont pas les même. Dans cette exemple, on remarque que les régions segmenté en bas sont plus rapproché lorsque on a augmenté ce paramètre.

- **Init_level_set**

Définit l'ensemble de niveaux de départ initial de l'algorithme.

Résultat pour une initialisation avec des vecteurs de 0

Résultat pour une initialisation avec des vecteurs de 1.

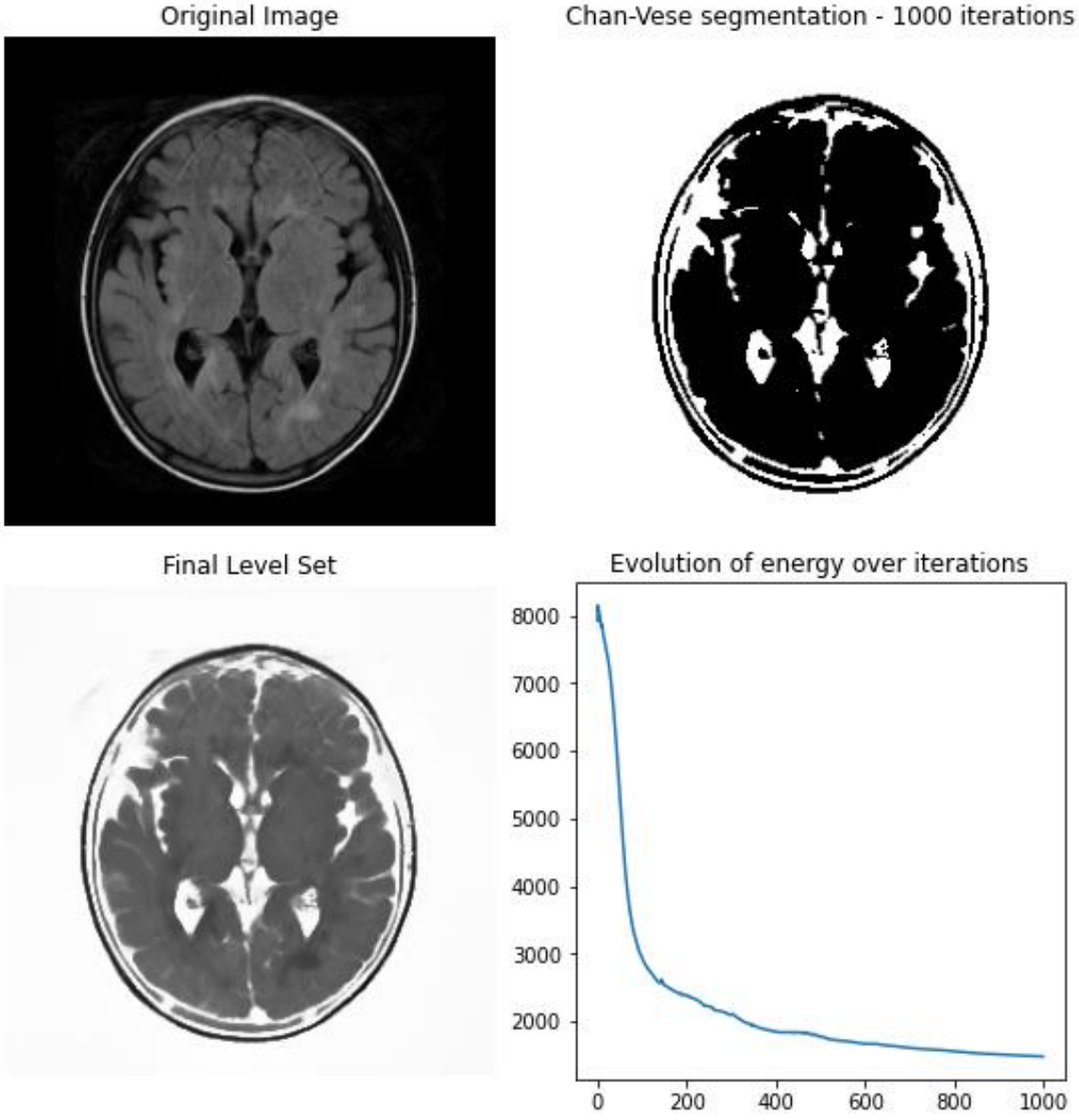
→L'algorithme de Chan et Vese est très sensible à l'initialisation. Pour une initialisation qui est totalement fausse, l'algorithme peut complétement diverger.

2.Segmentation

Dans cette partie, nous allons appliqué l'algorithme de chan vese pour la segmentation d'une image.

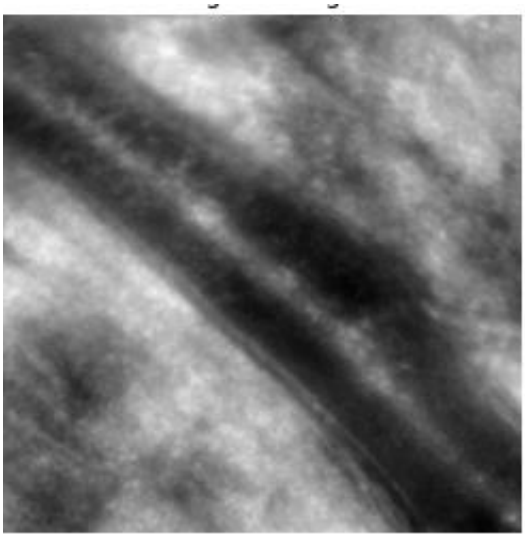
Tout d'abord , nous avons trouvé une difficulté pour trouver un bon compromis entre les différents paramètres.

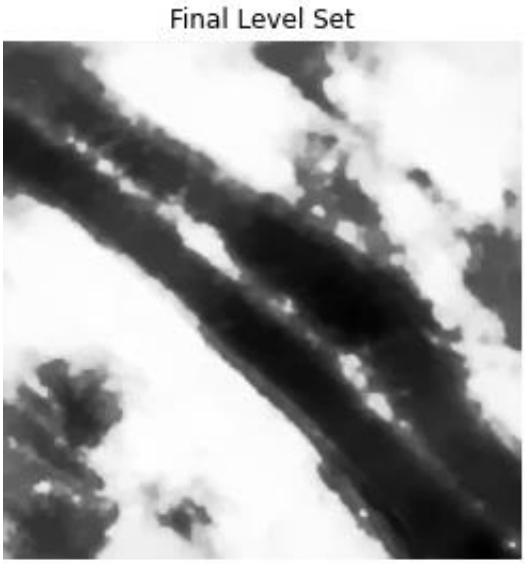
Nous allons travailler sur l'image de cerveau (brain2) et l'image de la rétine.

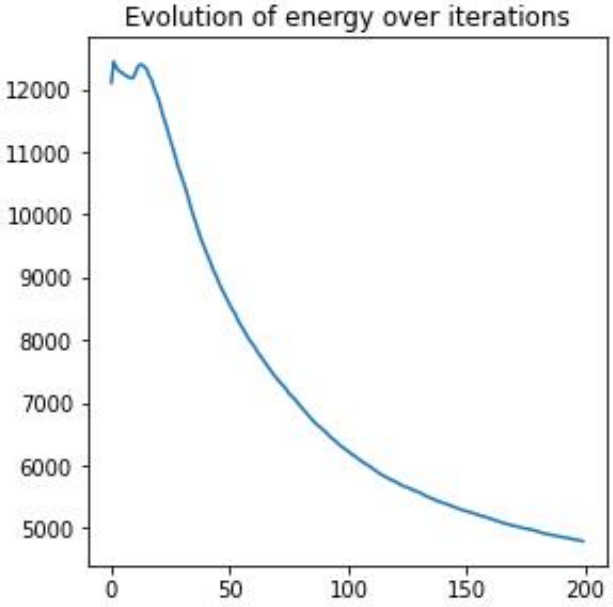



```
cv = chan_vese(image, mu=0.025, lambda1=5, lambda2=2, tol=1e-3, max_iter=1000,
               dt=0.5, init_level_set=init_ls, extended_output=True)

fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.flatten()
```








```
cv = chan_vese(image, mu=0.05, lambda1=5, lambda2=2, tol=1e-3, max_iter=200,
                dt=0.5, init_level_set=init_ls, extended_output=True)
```

Pour les deux images segmentées , on remarque que l'energie de l'image décroît en itérant.
Pour éliminer le bruit de niveau de gris et obtenir une meilleure segmentation l'image on peut faire des traitement morphologique sur l'image.