

IMA203-TP06

Leying ZHANG

January 2022

1 Méthode des contours actifs paramétriques

La définition de cette fonction est `active_contour(image, snake, alpha=0.01, beta=0.1, w_line=0, w_edge=1, gamma=0.01, max_px_move=1.0, max_num_iter=2500, convergence=0.1, *, boundary_condition='periodic', coordinates='rc')`

On va utiliser l'image "brain.bmp" pour expliquer chaque paramètres. Dans la Figure 1, on présente l'image originale de brain.

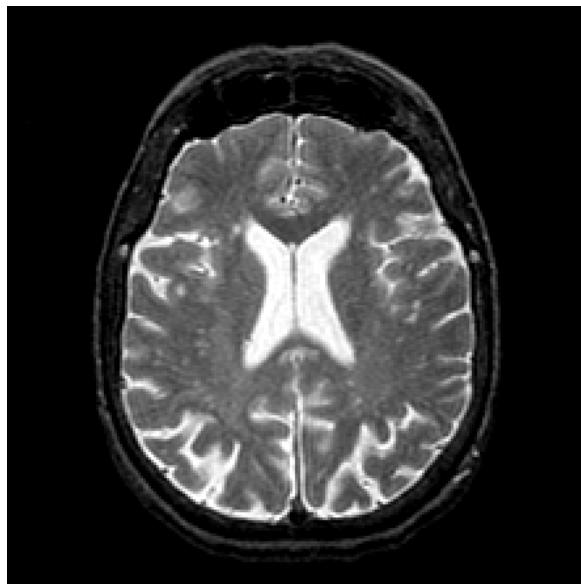


Figure 1: Image originale

1.1 Snake

D'après la définition, Le paramètre `snake` désigne les coordonnées initiales du snake. Pour les conditions limites périodiques, les points d'extrémité ne doivent pas être dupliqués.

Dans notre code, on peut changer la position initiale, et on trouve que cela a une influence très importante dans la segmentation, parce que on ne peut pas bien segmenter si la région est très petite, très grande ou la position est totalement mal choisie. Voir la Figure 2 .

Pour comparer l'effet de paramètres, tous les expériences ci-dessous utilisent $\alpha = 0.005, \beta = 0.01, w_edge = 1, w_line = 0, \gamma = 0.01, max_iterations = 2500$ comme les paramètres par défaut.

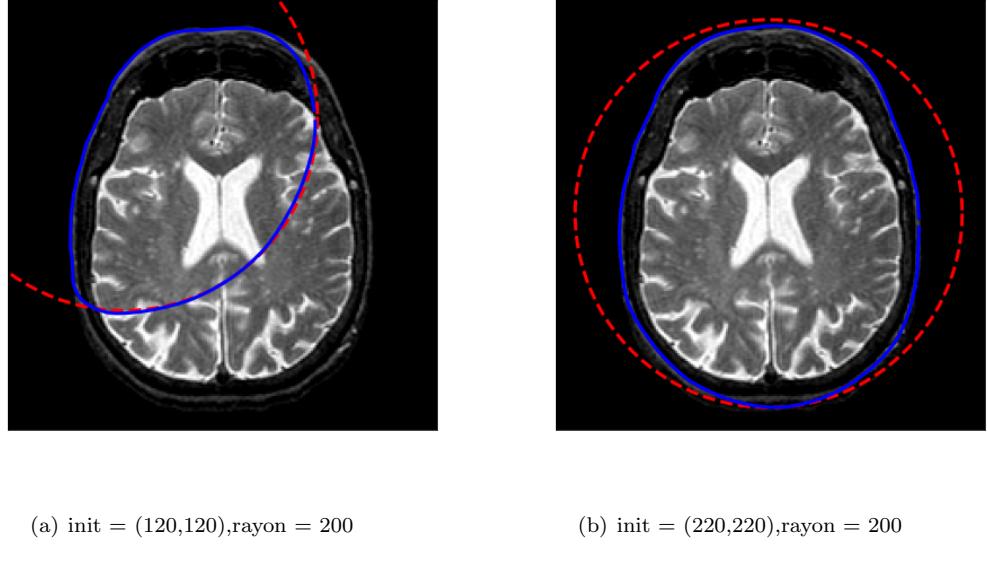


Figure 2: Segmentation de cerveaux avec initializations différentes

1.2 alpha

Selon la définition de cette fonction, alpha est le paramètre de forme de la longueur du serpent. Des valeurs plus élevées font que le serpent se contracte plus rapidement.

Dans la Figure 3, on trouve que quand alpha augmente, la cercle deviendra plus petite. En fait, alpha contrôle la tension de la courbe (tangente, la dérivée première) comme on a vu dans Eq.1

$$E_{interne} = \alpha(s) \left(\frac{dv}{ds} \right)^2 + \beta(s) \left(\frac{d^2v}{d^2s} \right)^2 \quad (1)$$

1.3 beta

β représente le paramètre de forme de lissage du snake. Des valeurs plus élevées rendent le serpent plus lisse.

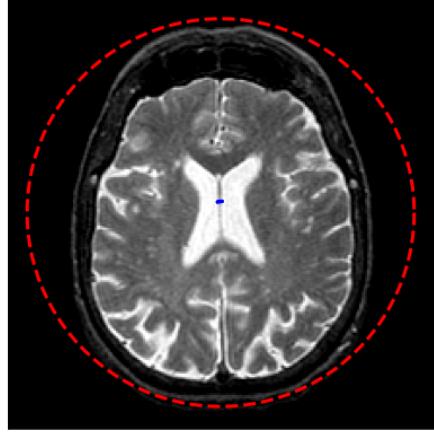
Dans Eq.1, on trouve aussi que β a une lien avec la dérivée seconde qui représente la courbure. Dans la Figure 4, on trouve aussi que quand β augmente, la courbe devient plus smooth, mais il est donc difficile de simuler le cerveau qui n'est pas une forme canonique.

1.4 w_edge

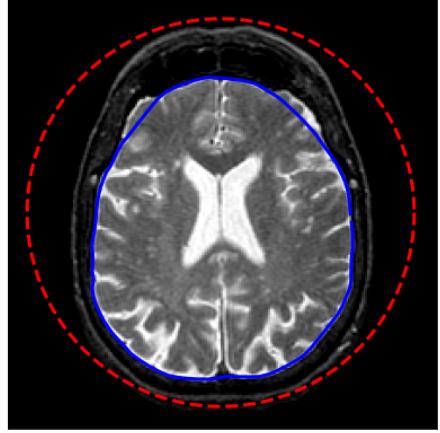
Ce paramètre contrôle l'attraction vers les bords. Utilisez des valeurs négatives pour repousser le snake des bords. En fait, ce paramètre intervient dans le calcul du gradient dans la fonction de contour actif, il s'agit donc d'une variable liée à l'image plutôt que de contrôler directement la forme de la courbe.

Pour mieux démontrer ce paramètre, nous avons repositionné l'initialisation du serpent à un endroit où le gradient est important, comme le bord du cerveau.

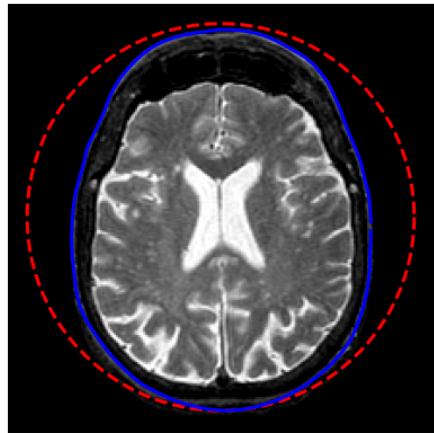
Dans la Figure 5, on observe que lorsque w_{edge} est un nombre positif, nos résultats de segmentation sont plus proches de l'endroit où le gradient est le plus grand, c'est-à-dire plus près du bord, et plus la valeur est grande, plus la proximité est prononcée. Lorsque w_{edge} est négatif, le résultat



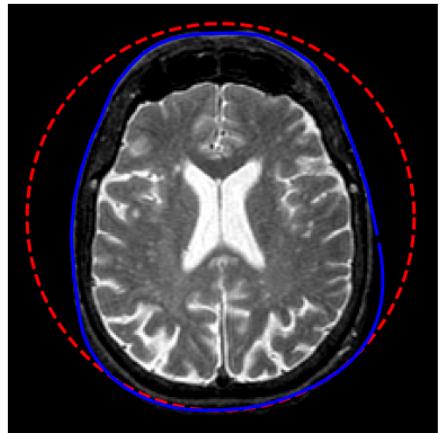
(a) $\alpha = 0.5$



(b) $\alpha = 0.05$

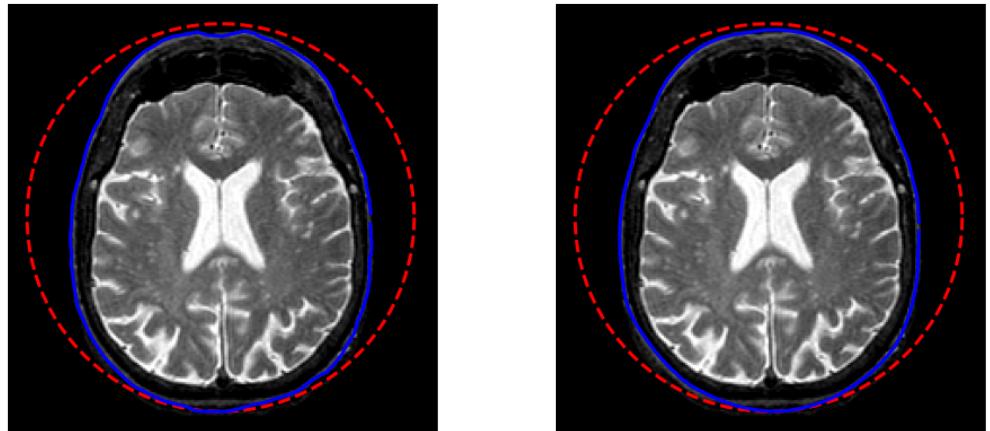


(c) $\alpha = 0.005$

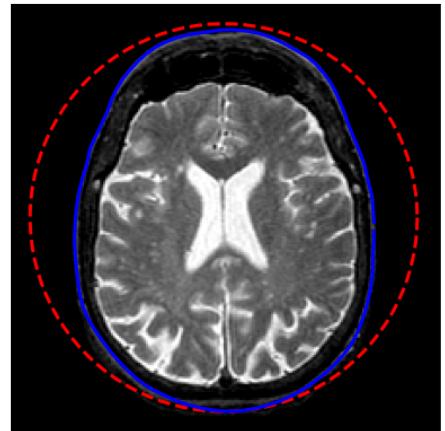


(d) $\alpha = 0.00005$

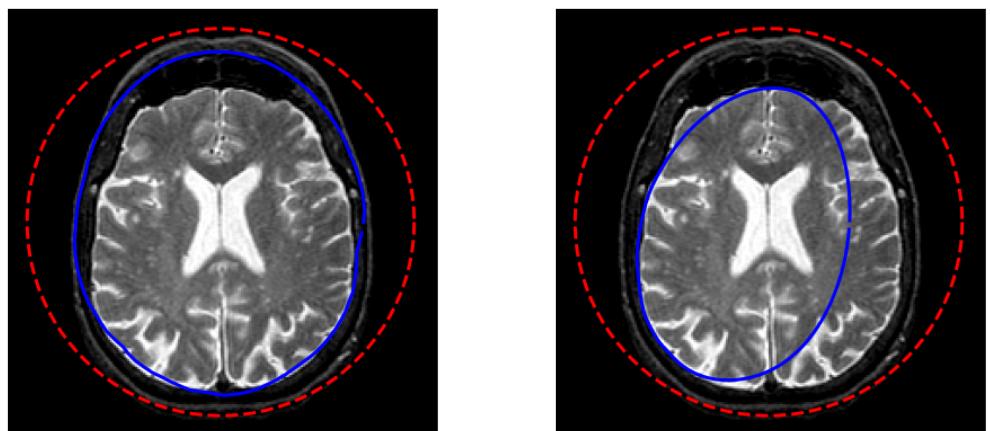
Figure 3: Segmentation de cerveaux avec α différentes



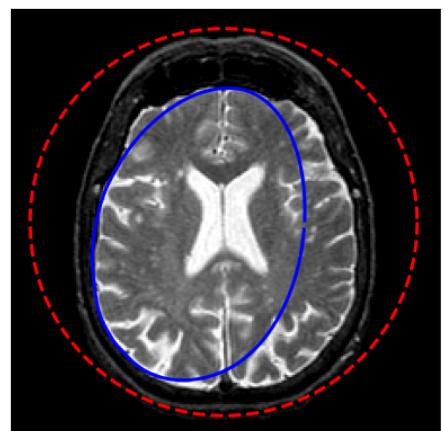
(a) $\beta = 0$



(b) $\beta = 0.1$



(c) $\beta = 3$



(d) $\beta = 10$

Figure 4: Segmentation de cerveaux avec β différentes

de notre segmentation est plus proche du bord, c'est-à-dire que plus la valeur absolue du nombre négatif est grande, plus il est éloigné du bord.

1.5 gamma

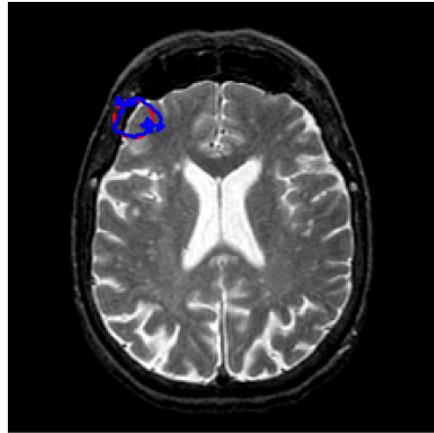
Selon la définition de la fonction, gamma signifie le paramètre de pas de temps explicite. D'après le cours, on trouve que dans l'évolution de la courbe dans le temps, le gamma est en fait l'inertie (Voir Eq. 2). Si cette inertie est très grande, la courbe va évoluer particulièrement, particulièrement lentement. Cette inertie aussi garantit qu'il n'y a pas de valeurs propres nulles ou petites lors de la recherche de la matrice inverse.

$$\tau \frac{\partial v}{\partial t} = -\alpha v'' + \beta v''' + F(v) \quad v(t+1) = (A + \tau I)^{-1}(F(v(t)) + \tau v(t)) \quad (2)$$

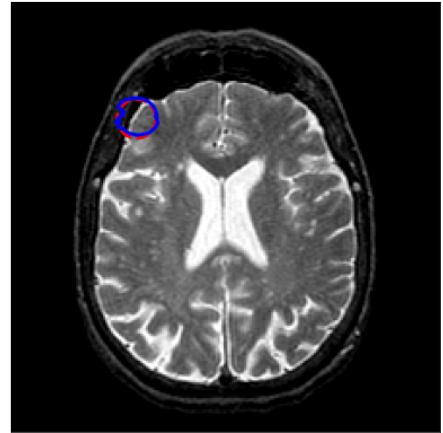
Dans la Figure 6, on trouve que les meilleurs résultats sont obtenus lorsque gamma = 0,01, qui est la valeur par défaut de la fonction de contour actif. Si gamma est trop grand, l'évolution de la courbe sera trop lente et le changement ne sera pas assez visible.

1.6 w_line

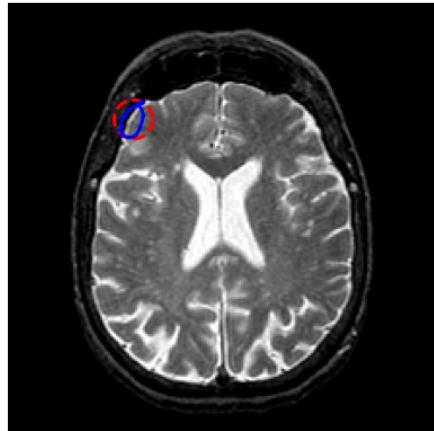
Ce paramètre, tout comme w_edge, contrôle l'attraction vers la luminosité. Utilisez des valeurs négatives pour attirer vers les régions sombres. Comme dans la Figure 7, si la valeur est plus élevée, la ligne bleue sera plus proche de la zone la plus lumineuse.



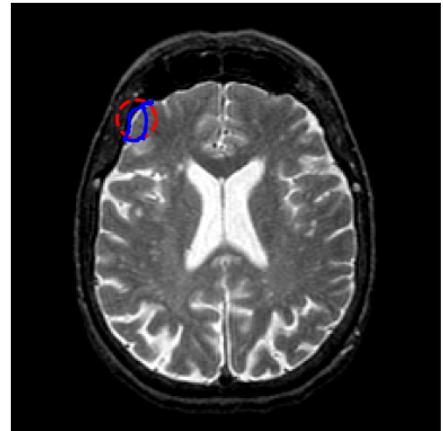
(a) $w_edge = -100$



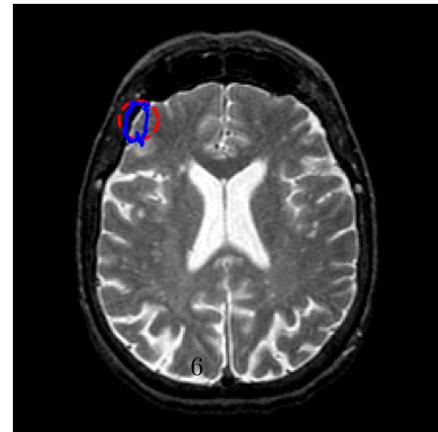
(b) $w_edge = -10$



(c) $w_edge = 1$

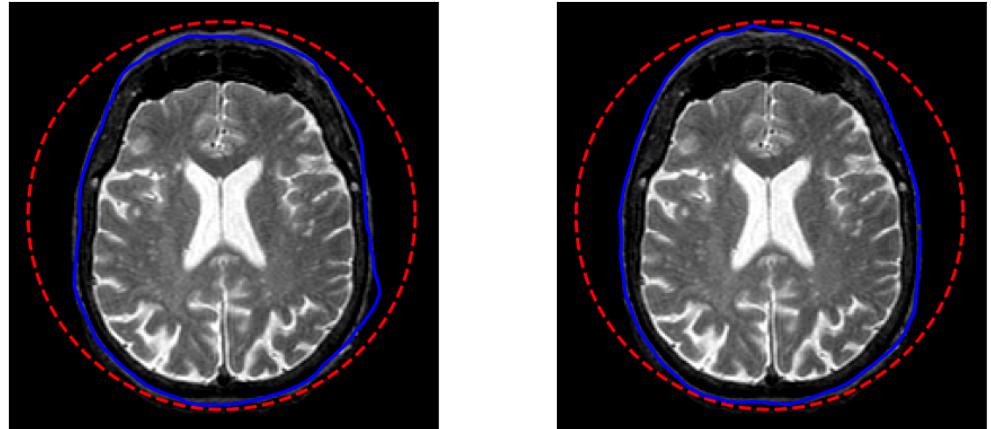


(d) $w_edge = 10$

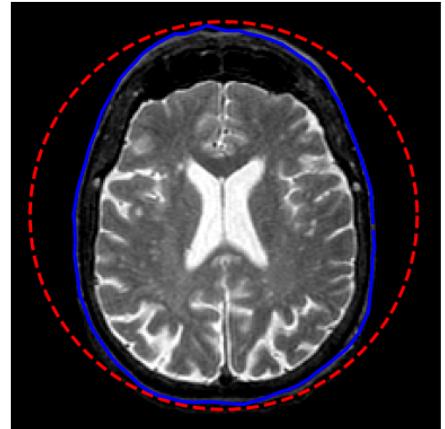


(e) $w_edge = 100$

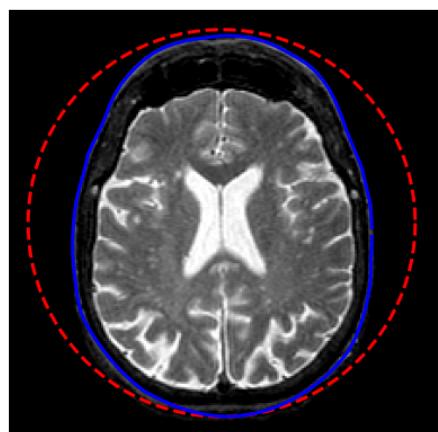
Figure 5: Segmentation de cerveaux avec w_edge différentes



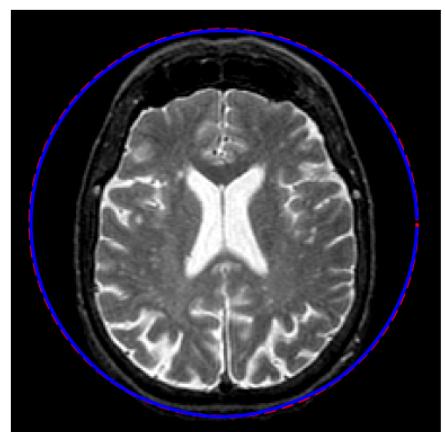
(a) $\text{gamma} = 0.0001$



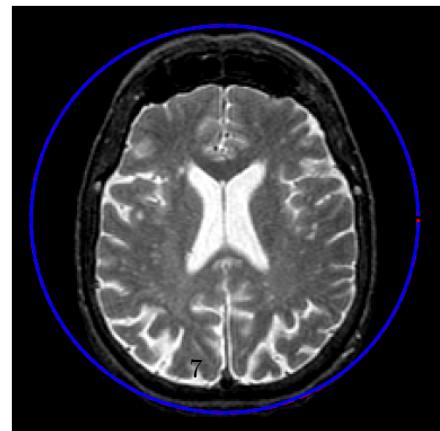
(b) $\text{gamma} = 0.001$



(c) $\text{gamma} = 0.01$

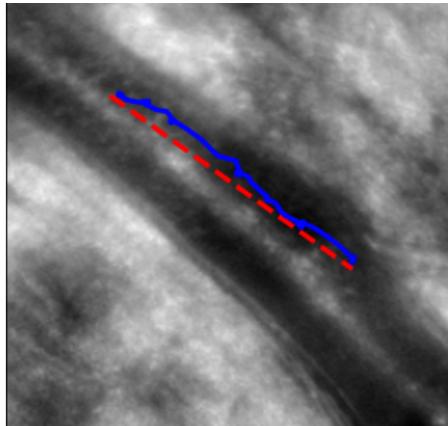


(d) $\text{gamma} = 0.1$

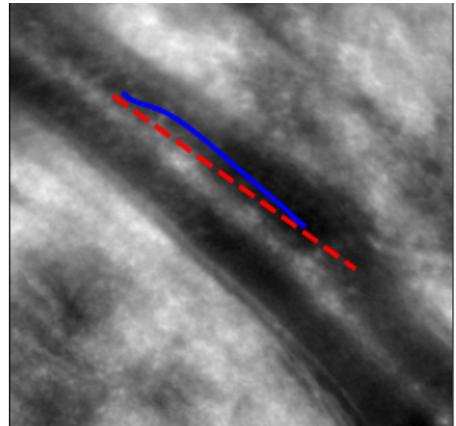


(e) $\text{gamma} = 1$

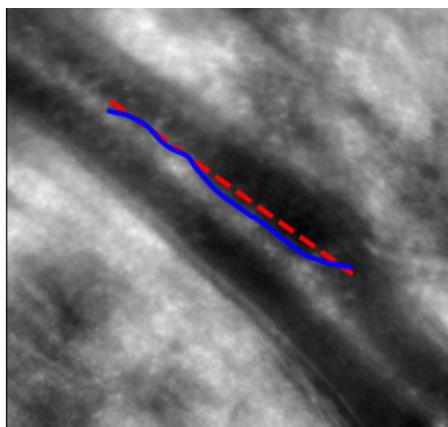
Figure 6: Segmentation de cerveaux avec gamma différentes



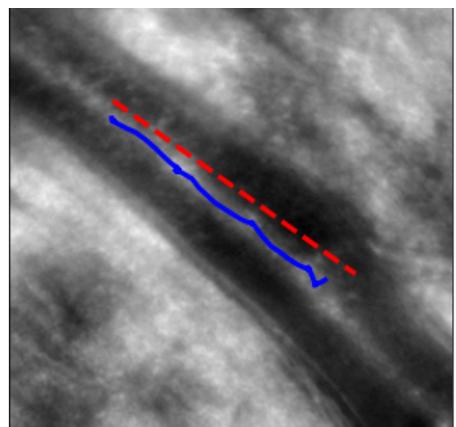
(a) $w_line = -100$



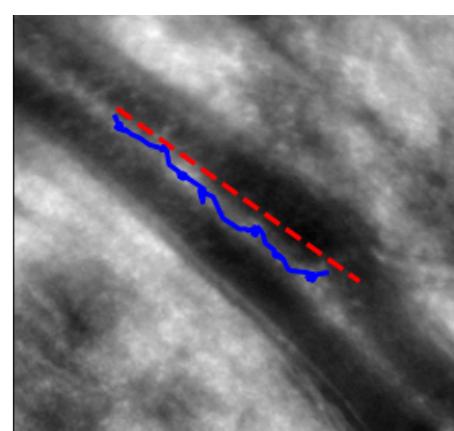
(b) $w_line = -10$



(c) $w_line = 0$



(d) $w_line = 10$



(e) $w_line = 100$

Figure 7: Segmentation de cerveaux avec w_line différentes

2 Méthode de Chan et Vese

Il s'agit d'un modèle de contour actif par évolution d'un ensemble de niveaux. Il peut être utilisé pour segmenter des objets sans frontières clairement définies.

Initialement, on a $amu = 0.25, lambda1 = 5, lambda2 = 1, tol = 1e - 3, max_iter = 200, dt = 0.5$, et on commence avec 8 cercles.

2.1 mu

Ce paramètre indique la longueur du bord. Des valeurs plus élevées produiront un bord "rond", tandis que des valeurs plus proches de zéro détecteront des objets plus petits. Dans la Figure 8, on trouve que le mu ne fonctionne pas bien quand il est particulièrement grand et quand il est particulièrement petit. Lorsque mu est particulièrement petit, il n'est pas assez long pour segmenter la partie principale du cerveau. Lorsque mu est particulièrement grand, il y a des bords peu clairs et l'énergie ne converge pas dans la même situation.

2.2 lambda1

Ce paramètre indique le paramètre de pondération "différence par rapport à la moyenne" pour la région de sortie ayant la valeur "True". S'il est inférieur à lambda2, cette région aura une plus grande plage de valeurs que l'autre. Les résultats sont de Figure 9.

2.3 lambda2

Ce paramètre indique le paramètre de pondération "différence par rapport à la moyenne" pour la région de sortie ayant la valeur "False". S'il est inférieur à lambda1, cette région aura une plus grande plage de valeurs que l'autre. Les résultats sont de Figure 10

2.4 tol

Tol représente la tolérance de variation de l'ensemble de niveaux entre les itérations. Si la différence de norme L2 entre les ensembles de niveaux des itérations successives, normalisée par la surface de l'image, est inférieure à la valeur de tol, l'algorithme considère que la solution a été atteinte. Dans la Figure 11, si tol est particulièrement petit, alors l'énergie s'arrête avant d'avoir convergé. Si tol est particulièrement grand, cela prendra beaucoup de temps et sera trop intensif en termes de calcul, à moins que cela ne soit nécessaire pour la précision.

2.5 max_iter

Il s'agit du nombre maximum d'itérations autorisées. Dans la Figure 12, si le nombre d'itérations est trop faible, l'arrêt sans convergence ne fonctionnera pas bien. Si le nombre d'itérations est trop important, le calcul sera très lourd.

2.6 dt

dt est un facteur de multiplication appliqué aux calculs pour chaque étape. Il sert à accélérer l'algorithme. Dans la figure 13, on constate qu'un dt trop grand ou trop petit ne fonctionne pas bien. Si dt est trop petit, la vitesse sera très lente, et si dt est trop grand, le minimum sera manqué et la convergence ne sera pas bonne.

2.7 circleNum

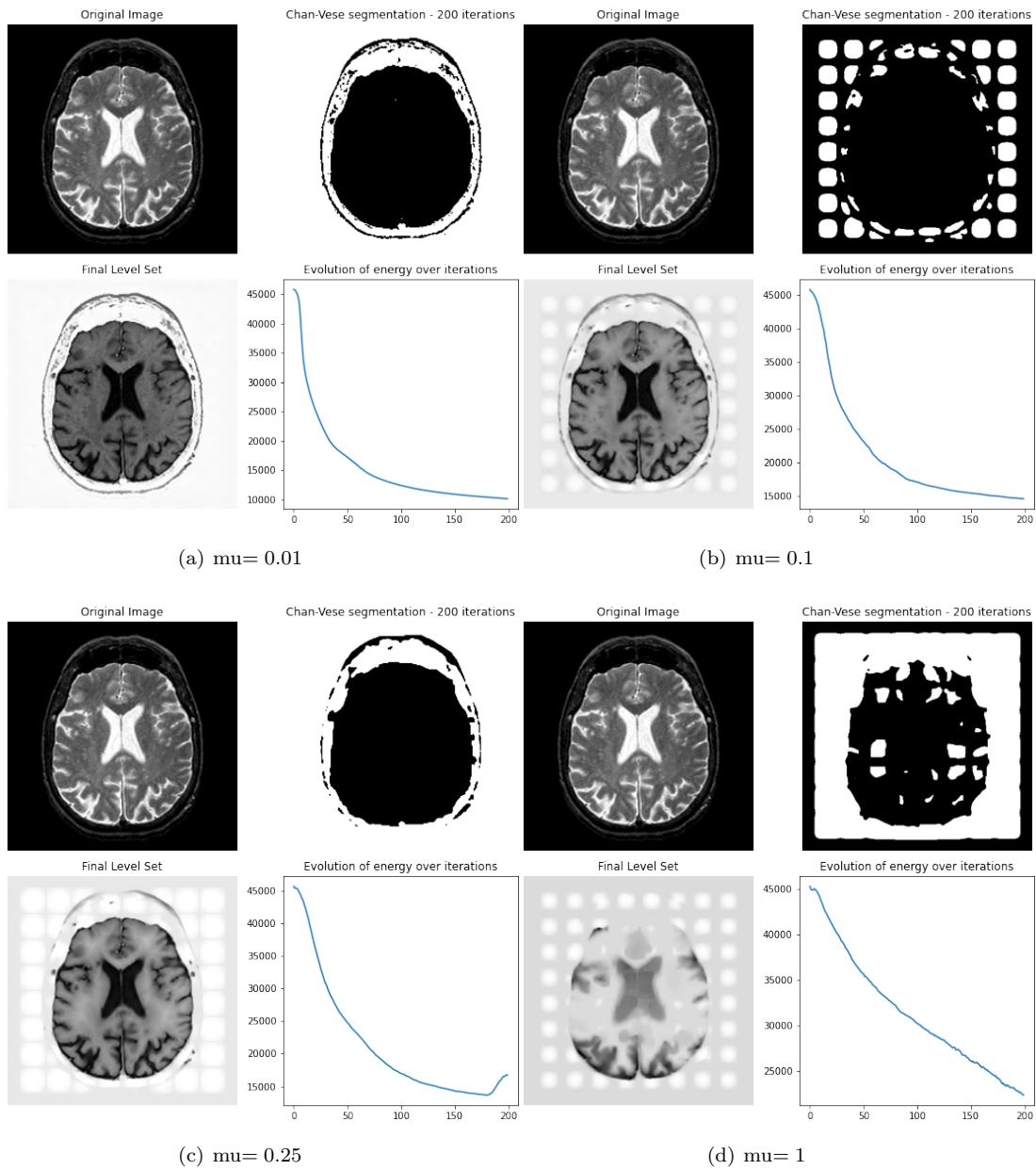


Figure 8: Segmentation de cerveaux avec μ différentes

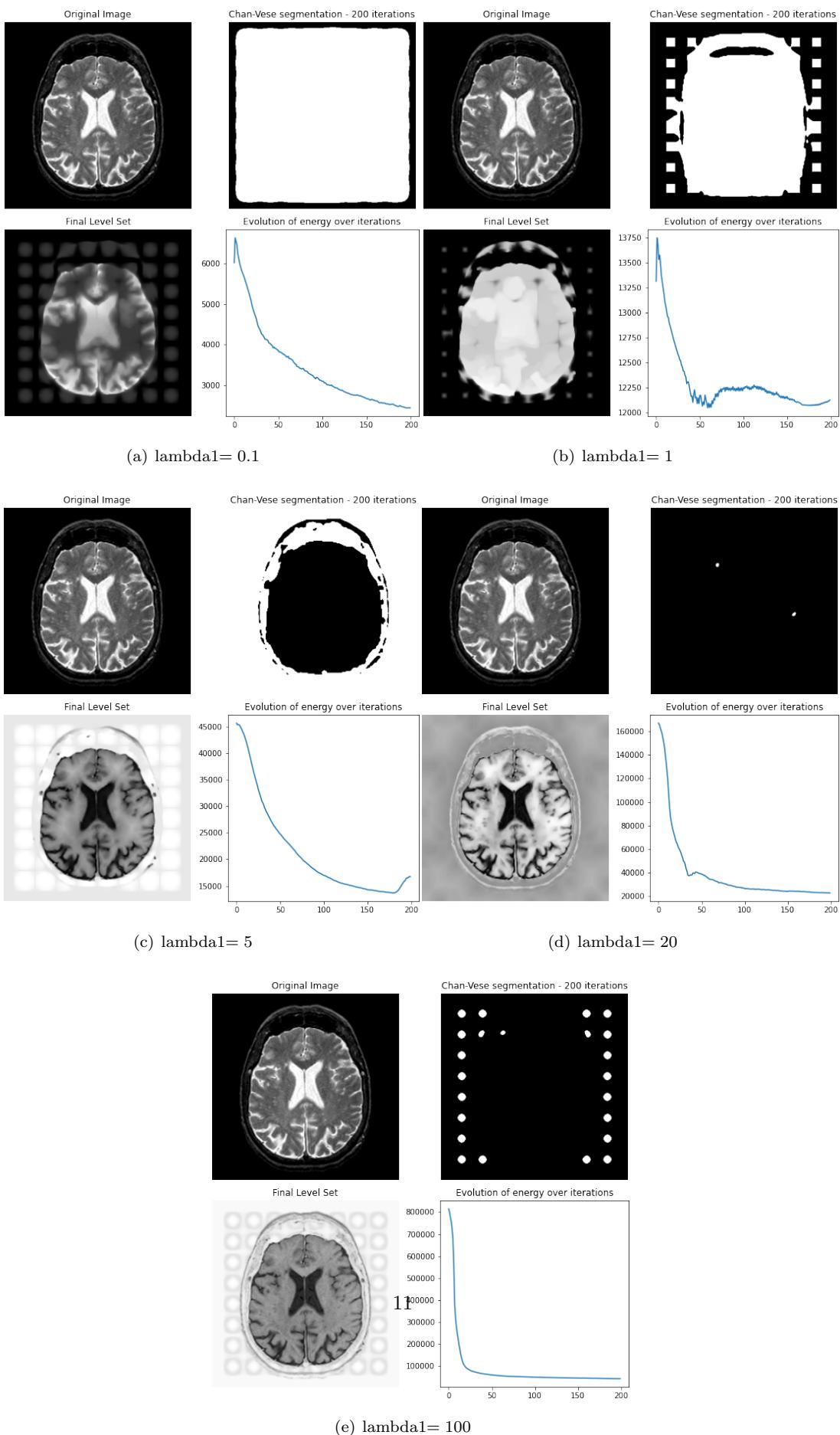
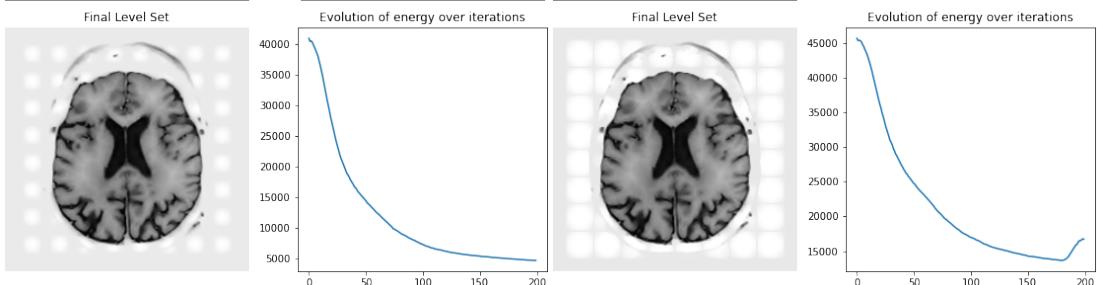
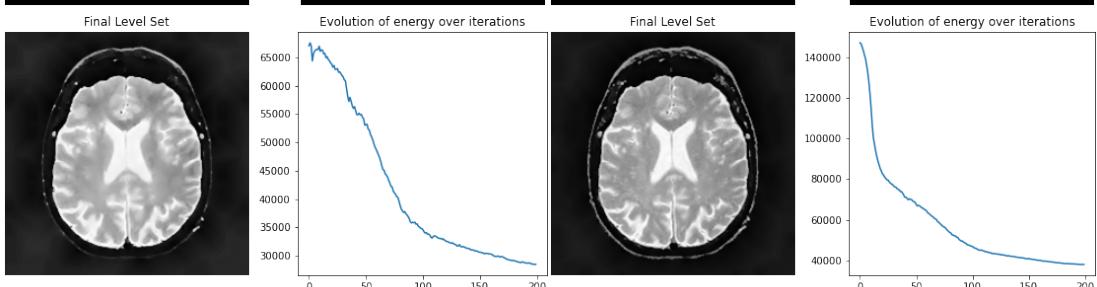
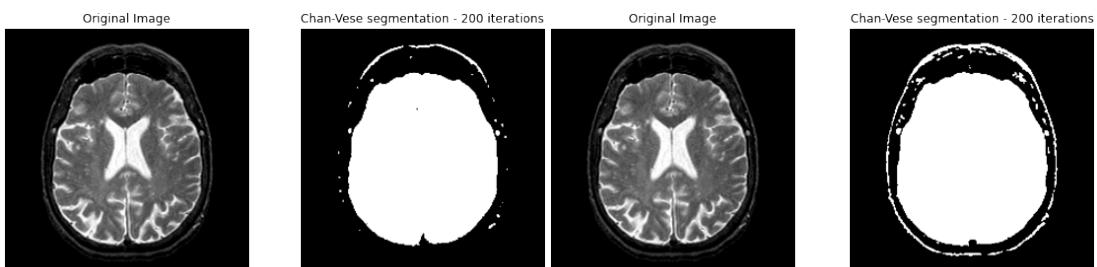


Figure 9: Segmentation de cerveaux avec λ_1 différentes



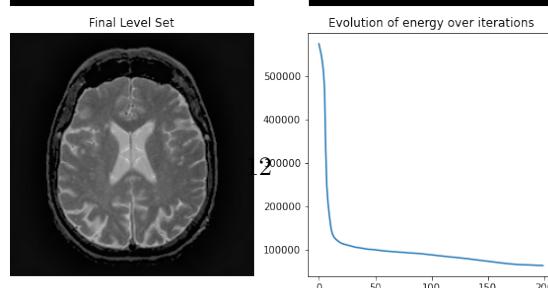
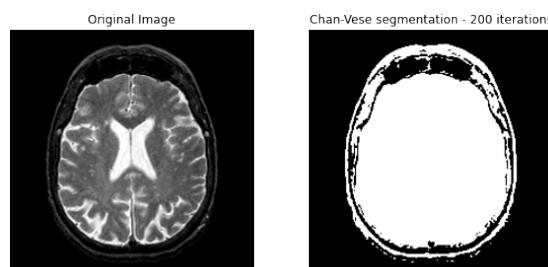
(a) $\lambda_2 = 0.1$

(b) $\lambda_2 = 1$



(c) $\lambda_2 = 5$

(d) $\lambda_2 = 20$



(e) $\lambda_2 = 100$

Figure 10: Segmentation de cerveaux avec λ_2 différentes

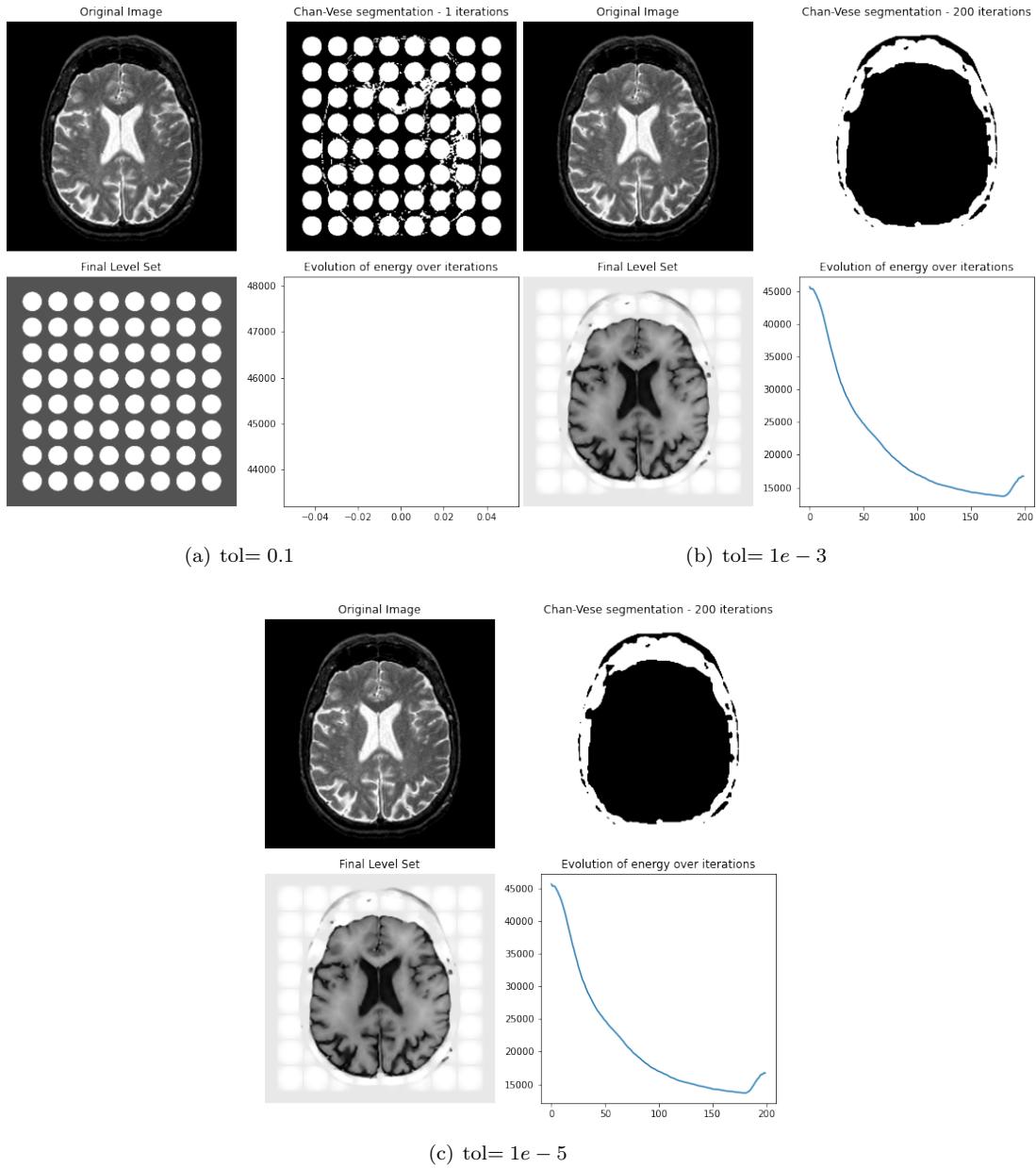


Figure 11: Segmentation de cerveaux avec λ_2 différentes

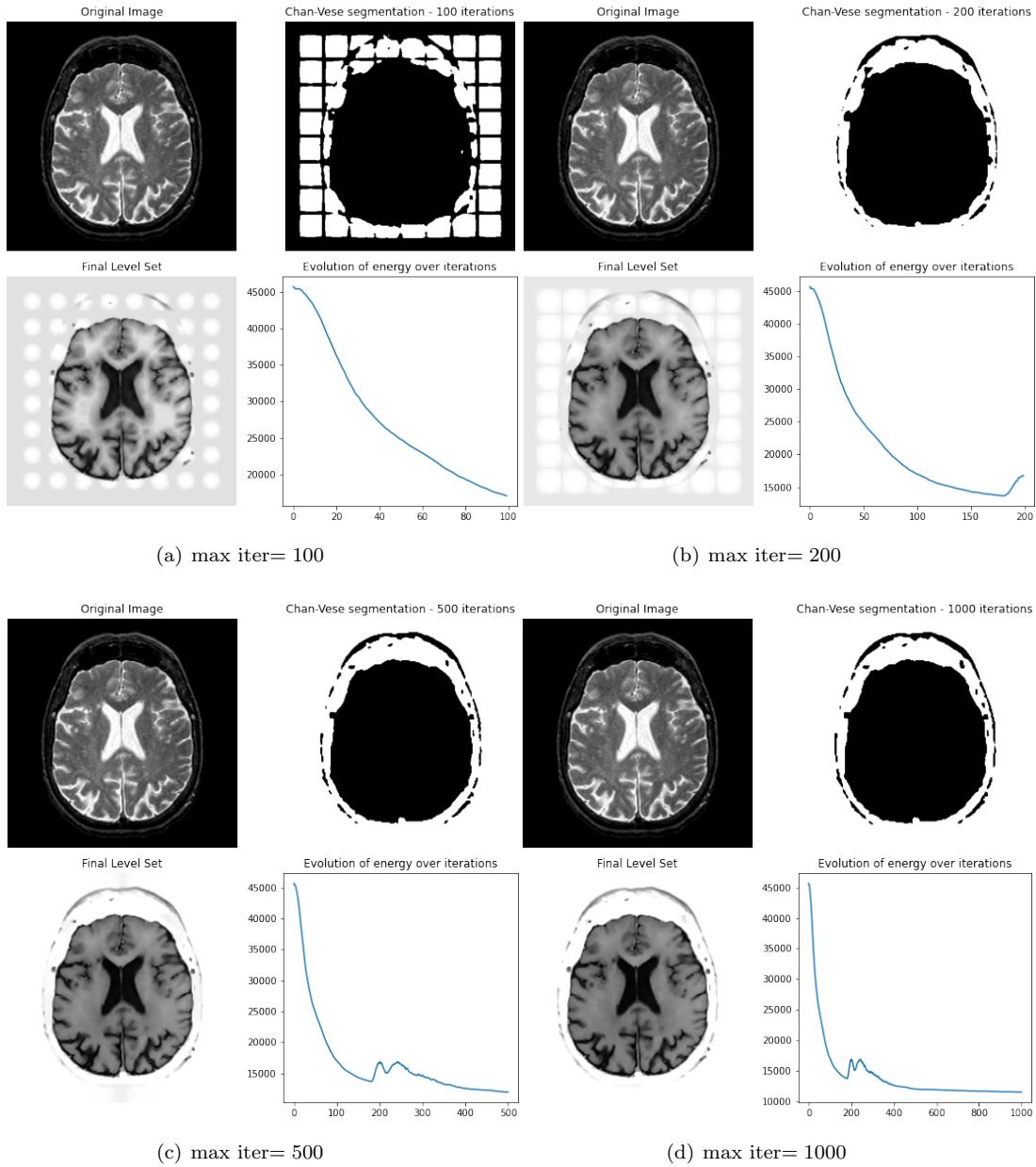


Figure 12: Segmentation de cerveaux avec max_iter différentes

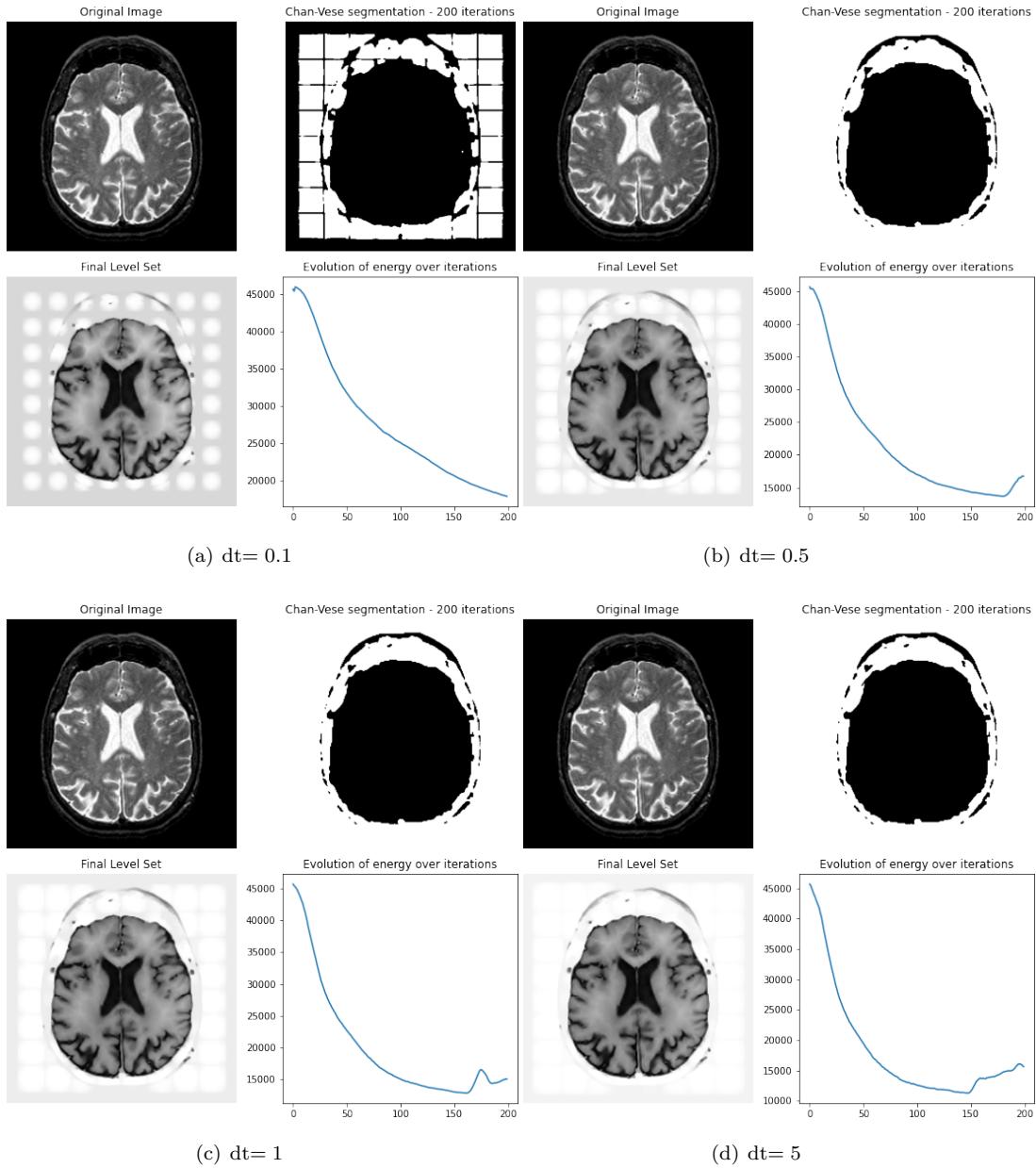


Figure 13: Segmentation de cerveaux avec dt différentes

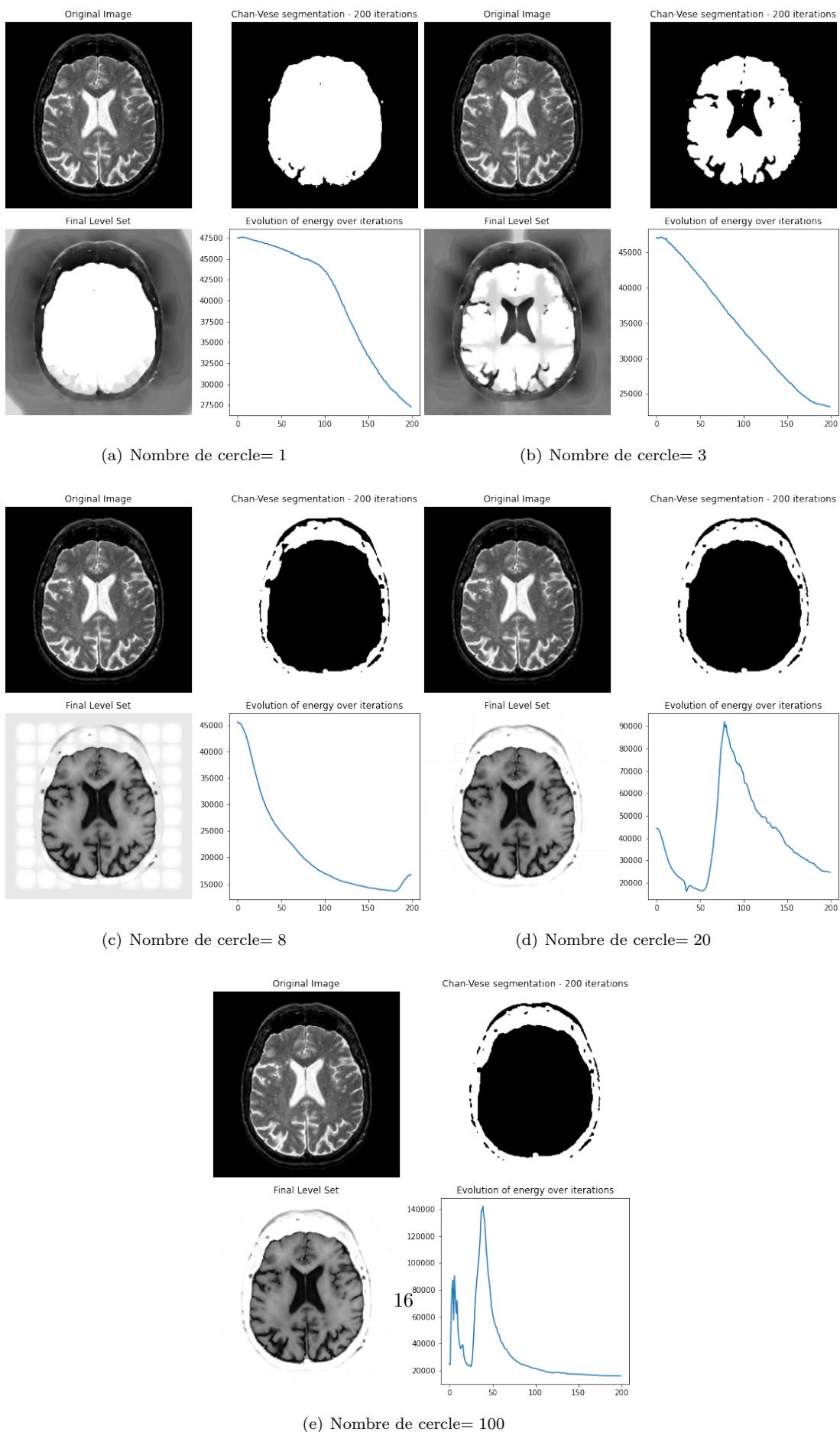


Figure 14: Segmentation de cerveaux avec dt différentes

3 Segmentation

3.1 Méthode des contours actifs paramétriques

Pour l'image originale, on peut utiliser les paramètres $\alpha = 0.005, \beta = 0.01, w_edge = 1, w_line = 0, \gamma = 0.01, max_iterations = 2500$ et l'initialisation est un cercle de (220,220) de rayon 200 pour segmenter le cerveaux dans la figure 15. On a choisi ces paramètres parce qu'ils nous ont semblé donner les meilleurs résultats après les tentatives mentionnées ci-dessus, tant en ce qui concerne l'effet de alpha et bêta sur la snake et l'effet des autres paramètres sur le traitement de l'image.

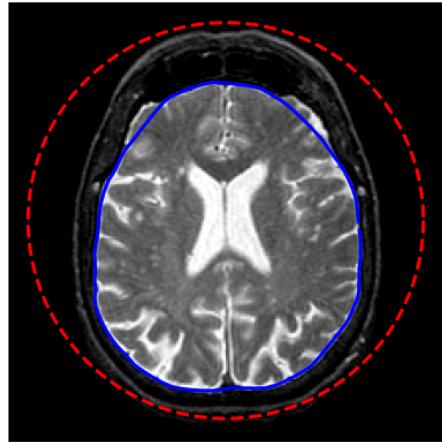


Figure 15: Resultat de méthode 1

On a voulu ajuster à nouveau les paramètres pour diviser la partie centrale du cerveau. Mais c'était difficile, car cette méthode est sensible aux paramètres, et après de nombreuses expériences, les résultats n'étaient toujours pas très bons dans la Figure 16. Les paramètres sont $\alpha = 0.005, \beta = 1, w_edge = 2, w_line = 0, \gamma = 0.005, max_iterations = 2500$ et l'initialisation est un cercle de (220,220) de rayon 50.

3.2 Méthode de Chan et Vese

Pour segmenter le cerveaux, on choisit $mu = 0.25, lambda1 = 5, lambda2 = 1, tol = 1e - 3, max_iter = 200, dt = 0.5$, et on commence avec 8 cercles. Le resultat est dans la figure 17. Il faut choisir la bonne itération pour nous assurer qu'elle est convergente. Les paramètres sont également choisis sur la base des essais de la section 2.

Toutefois, si c'est la partie centrale du cerveau que nous voulons diviser, un tel choix de paramètres n'aurait pas de sens. Nous avons donc procédé à des ajustements. On a essayé et on choisi de n'utiliser qu'un seul cercle car nous savons clairement quelle est la position de l'objet à segmenter. En essayant cela, on constate que si le mu est petit, il y aura beaucoup de points errants, et en plus de la division du milieu, il y aura d'autres points errants supplémentaires. Finalement, on choisit $m = 0.8, lambda1 = 20, lambda2 = 1, tol = 1e - 3, max_iteration = 300$ et le résultat est de la figure 18.

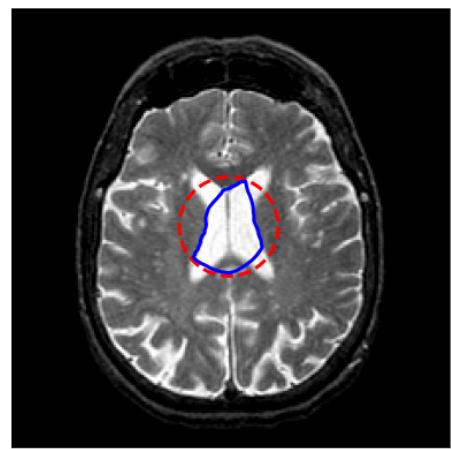


Figure 16: Centre de cerveaux:Résultat de méthode 1

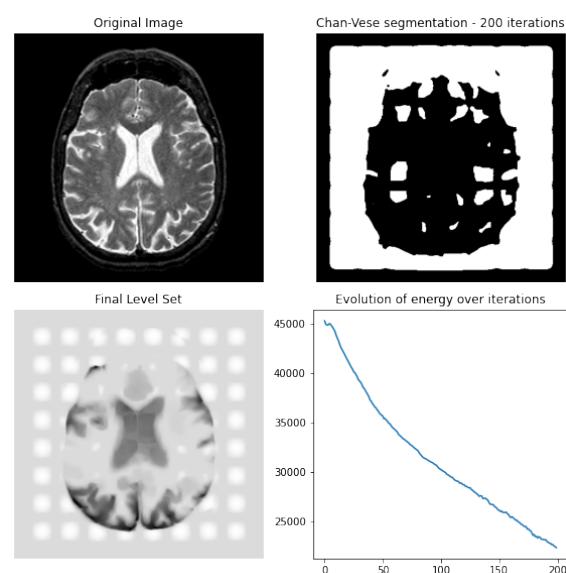


Figure 17: Resultat de méthode 1

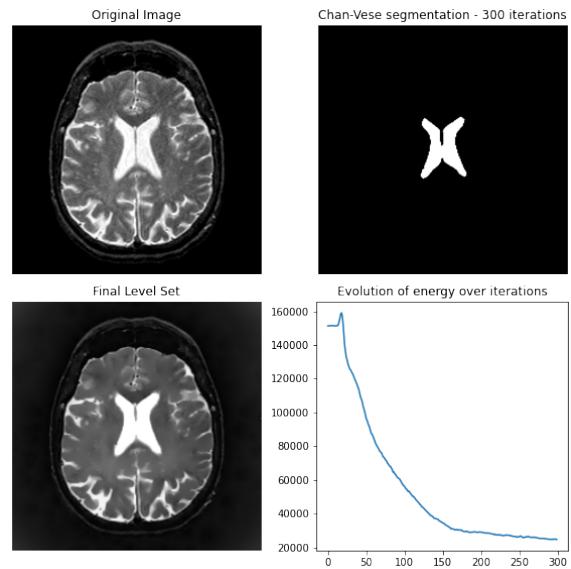


Figure 18: Centre de cerveaux: Résultat de méthode 2

3.3 Comparaison et amélioration

Je pense que la deuxième méthode est plus facile à comprendre et que les paramètres sont mieux ajustés et plus spécifiques.

Je pense que pour la première méthode, nous devons encore ajuster ses paramètres, notamment pour w_{edge} , car je pense que c'est un paramètre extrêmement important pour l'attractivité des bords, c'est-à-dire lors du calcul du gradient de l'image pendant le calcul.