SEMI-AUTOMATIC BRAIN TUMOR SEGMENTATION ALGORITHM

Shreyas Katdare(22B0636) & Sanskar Shaurya (22B0985)

May 2, 2024

1. INTRODUCTION

- Brain tumors and normal tissues share similar gray levels, making segmentation difficult.
- Tumor regions vary in intensity, leading to isolated holes in segmentation maps.
- Many image segmentation algorithms exist, but applying them to brain tumor segmentation is challenging.
- In order to solve this problem, a new semi-automatic segmentation algorithm dedicated to brain tumors is proposed.

Outline of the proposed Algorithm

- Step 1 : Otsu based N-level thresholding algorithm is used to segment the original MRI image into (N+1) classes.
- In order to reduce the influence of noises or weak edges on segmentation maps, an edge-aware filter is utilized to smooth the MRI image, and the Otsu based thresholding algorithm runs on the smoothed image as well.
- The two segmentation maps are fused with the rule of K Nearest Neighbors (KNN).
- Step 2 : a seed point needs to be manually placed on each tumor region by the user. The final segmented tumor regions are extracted using a bi- directional region growing algorithm.

2.1 Multiscale image representation

Given an image g, we seek a piecewise smooth base layer s, capturing large scale variations in intensity, and a residual detail layer d, containing the smaller scale details in the image.

$$\hat{s} = \arg\min_{s} \left\{ \sum_{\mathbf{p}} \left((s(\mathbf{p}) - g(\mathbf{p}))^2 \right) \right.$$
$$\left. + \lambda \left(a_{g,x}(\mathbf{p}) \cdot \left(\left(\frac{\partial s}{\partial x} \right) (\mathbf{p}) \right)^2 + a_{g,y}(\mathbf{p}) \cdot \left(\left(\frac{\partial s}{\partial y} \right) (\mathbf{p}) \right)^2 \right) \right\} \quad (1)$$

where $\hat{s}$ refers to the optimal smooth base layer, and $\mathbf{p}$ denotes the spatial location of a pixel. The first item $(s(\mathbf{p}) - g(\mathbf{p}))^2$ is to minimize the difference between s and g; and the second item is to minimize the vertical and horizontal gradients except where the underlying image has significant gradients in its log-luminance channel L of image g [12]. The parameter λ balances the relative weight of the two terms. $a_{g,x}$ and $a_{g,y}$ denote smoothness weights which are related to vertical and horizontal gradients of L, respectively.

2.1 Multiscale image representation (Contd.)

The smoothness requirement is that the smooth base layer s should preserve the boundary as it is and smooth other regions as much as possible. Hence, $a_{g,x}$ (or $a_{g,y}$) should be inverse proportional to vertical (or horizontal) gradient of L. Here, the two smoothness weights are defined in the same manner as in [11, 12].

$$a_{g,x}(\mathbf{p}) = \left(\left| \frac{\partial L}{\partial x}(\mathbf{p}) \right|^{\alpha} + \varepsilon \right)^{-1} \tag{2}$$

and

$$a_{g,y}(\mathbf{p}) = \left(\left| \frac{\partial L}{\partial y}(\mathbf{p}) \right|^{\alpha} + \varepsilon \right)^{-1} \tag{3}$$

where ε is a small positive constant, and the parameter α controls the sensitivity to the gradients of g. Typically, $\varepsilon = 0.0001$ and $\alpha = 1$.

2.1 Multiscale image representation (Contd.)

$$\hat{s} = \arg\min_{s}\{(s-g)^T(s-g) + \lambda\left(s^T D_x^T A_x D_x s + s^T D_y^T A_y D_y s\right)\} \tag{4}$$

where D_x and D_y are discrete differentiation operators, A_x and A_y are diagonal matrices containing the smoothness weights $a_{g,x}$ and $a_{g,y}$, respectively. The vector s that minimizes Eq.(4) is uniquely defined as the solution of the linear system

$$(I + \lambda L_g)\, s = g \tag{5}$$

where $L_g = D_x^T A_x D_x + D_y^T A_y D_y$.
Eq. (5) tells us that s is obtained from g by applying a nonlinear operator F_λ, which depends on g :

$$s = F_\lambda(g) = (I + \lambda L_g)^{-1}\, g \tag{6}$$

2.1 Multiscale image representation(Contd.)

Since this operator is spatially variant, it is hard to analyze its frequency response. We assume $a_{g,x} = a_{g,y} = a$, then Eq.(6) is simplified by

$$s = F_\lambda(g) \approx (I + \lambda a L)^{-1} g \tag{7}$$

The residual detail layer d can be obtained by subtracting s from the original image g, namely

$$d = g - s \tag{8}$$

2.2 Multi-scale Otsu based segmentation

- The multilevel Otsu algorithm on M x N image works as follows:
  - Let $\{C_1, C_2, \ldots, C_{N_c}\}$ denote the N_c classes segmented by the threshold set $\{T_1, T_2, \ldots, T_{N_c}\}$.
  - Let ω_i be the percentage of pixels belonging to class C_i, calculated as:

    $$\omega_i = \frac{\sum_{j=T_{i-1}+1}^{T_i} n_j}{M \times N},$$

    where n_j represents the number of pixels with gray level j.
  - μ_i refers to the mean of class C_i, calculated as:

    $$\mu_i = \frac{\sum_{j=T_{i-1}+1}^{T_i} j \cdot n_j}{\sum_{j=T_{i-1}+1}^{T_i} n_j}$$

  - The between-class variance of the image I is defined as:

    $$\sigma^2 = \sum_{i=1}^{N} \omega_i (\mu - \mu_i)^2,$$

2.2 Multi-scale Otsu based segmentation (Contd.)

- where μ denotes the mean of the whole image, calculated as: $\mu = \sum_{j=0}^{j=255} \frac{j \cdot n_j}{M \times N}$
- The optimal threshold set T should be the one with the maximum between-class variance, i.e., $\arg\max_T \sigma^2$.
- In subsection 2.1, we can get a smoothed version of the original image I, which is denoted as I'.
- We apply this algorithm on both the images
  - First, find the optimal threshold sets $\widehat{T}$ and $\widehat{T}'$ for I and I' respectively using multilevel Otsu algorithm.
  - The original image I and its smoothed version I' are segmented into N classes respectively, namely $I \xrightarrow{\widehat{T}} C = \{C_1, \ldots, C_N\}$, and $I' \xrightarrow{\widehat{T}'} C' = \{C'_1, \ldots, C'_N\}$.
  - In most cases, C is not identical to C', which means that some pixels are mapped into different classes in C and C'.

2.2 Multi-scale Otsu based segmentation (Contd.)

- For these controversial pixels, we need to reconsider their class labels. Here, K Nearest Neighbor (KNN) is used to solve this problem.
- For a pixel $\mathbf{p}$ satisfying $\mathbf{p} \in C_i \wedge \mathbf{p} \in C'_j \wedge i \neq j, (1 \leq i, j \leq N)$, we find its neighborhood set $H_\mathbf{p}$ with the size of K where any element $\mathbf{q}$ satisfying $\mathbf{q} \in C_l \wedge \mathbf{q} \in C'_l, (l = i \text{ or } l = j)$, if the number of the elements belonging to C_i is larger than the number of those belonging to C'_j, $\mathbf{p}$ is classified into C_i; otherwise, $\mathbf{p}$ is mapped into C'_j, namely

$$\mathbf{p} \in \begin{cases} C_i, & \text{if } \sum_{\mathbf{q} \in H_\mathbf{p}} \mathrm{Id}\left(\mathbf{q} \in C_i\right) > \sum_{\mathbf{q} \in H_\mathbf{p}} \mathrm{Id}\left(\mathbf{q} \in C'_j\right) \\ C'_j, & \text{otherwise} \end{cases} \tag{11}$$

where $\mathrm{Id}(*)$ is an indicator function which is defined as

$$\mathrm{Id}(*) = \begin{cases} 1, & * = \text{ true} \\ 0, & * = \text{ false} \end{cases} \tag{12}$$

2.3 Bi-directional region growing

- Our algorithm contains twice growing: the first one with the initial seeds located in tumor regions is to find the rough interest region; and the second one where one seed is selected out of the rough ROI is to eliminate the isolated holes in segmentation map.
- Let $\bar{C}$ denote the segmentation map obtained in Subsection 2.2, the bi-directional Region growing algorithm is defined as follows,
- Inner-to-boundary region growing: The initial seed $\mathbf{p}$ is located in the tumor region, $\mathbf{q}$ is one element of $\mathbf{p}$'s neighboring pixel set, and $\bar{C}(\mathbf{p})$ and $\bar{C}(\mathbf{q})$ denote the class labels of $\mathbf{p}$ and $\mathbf{q}$, respectively. If $|\bar{C}(\mathbf{p}) - \bar{C}(\mathbf{q})| < \lambda, \mathbf{p}$ and $\mathbf{q}$ belong to the same region. Otherwise, $\mathbf{p}$ and $\mathbf{q}$ belong to different regions. Region is grown from the seed pixel $\mathbf{p}$ by adding in neighboring pixels that are similar. This whole process is continued until the region does not grow any more.

2.3 Bi-directional region growing (Contd.)

- The segmentation result is denoted by *IRG* after this region growing. The regions that include initial seeds are marked as " 1 ", other regions are marked as " 0 ".
- Outer-to-boundary region growing: The initial seed **p** is located out of the tumor region, and the threshold is set to be 1 . Run region growing algorithm and get the segmentation result which is denoted by *ORG* after region finishing growing. The region that includes initial seed is marked as " 1 ", other regions are marked as " 0 ".
- We have used simple BFS graph traversal algorithm to achieve this.

Conclusion

- The proposed semi-automatic brain tumor segmentation algorithm:
  - Utilizes Otsu based N-level thresholding algorithm.
  - Incorporates an edge-aware filter for noise reduction.
  - Employs K Nearest Neighbors (KNN) for improved segmentation.
  - Requires manual seed placement followed by bi-directional region growing.

Evaluation Metric

In order to validate the performance of the proposed segmentation algorithm quantitatively, we use an objective measure which is constructed as follows : Let S and T denote the automatic and ground-truth segmentation map of an MRI image, respectively. Both of them are binary maps, where each element 1 corresponds to a pixel in tumor region and 0 does a pixel in formal region. For a pixel $\mathbf{p}$, if $T(\mathbf{p}) = 1$ and $S(\mathbf{p}) = 1$, it is a true positive; for another pixel $\mathbf{q}$, if $T(\mathbf{q}) = 0$ and $S(\mathbf{q}) = 1$, it is a false negative. Based on them, two measures are achieved, namely true positive rate (tpr) and false positive rate (fpr). The former one is the percentage of true positive accounting for the total positives and the latter one is the percentage of false positive accounting for the total negatives:

$$tpr = \frac{\sum_{\mathbf{p}}(S\&T)_{\mathbf{p}}}{\sum_{\mathbf{p}}(T)_{\mathbf{p}}} \tag{13}$$

and

Evaluation Metric (Contd.)

$$fpr = \frac{\sum_{\mathbf{p}} (S \& (1 - T))_{\mathbf{p}}}{\sum_{\mathbf{p}} (1 - T)_{\mathbf{p}}} \tag{14}$$

A good segmentation should maximize true positive rate, meanwhile minimize the false positive rate. If the two rates are plotted on a graph's x-axis and y-axis, the point $(1, 0)$ corresponds to the perfect segmentation result: S and T are identical. Each point on the graph corresponds to a segmentation result. Based on its distance from the perfect point, the quantitative evaluation measure is defined as

$$Q = 1 - \sqrt{\frac{(fpr)^2 + (1 - tpr)^2}{2}} \tag{15}$$

Here, Q falls in the interval of $[0, 1]$ where 1 refers to the perfect segmentation result and 0 does the worst segmentation result.

Final Results :

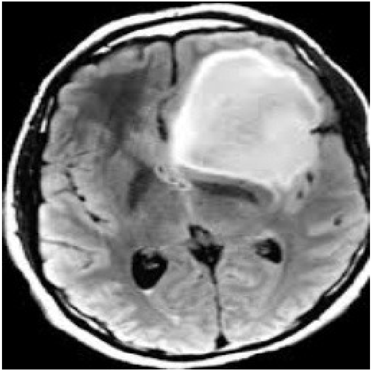

Figure: Input

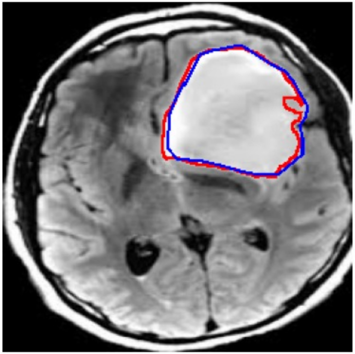

Figure: Red line: Outline given by algorithm
Blue line: Ground truth outline

```
True Positive Rate:  0.9552572706935123
False Positive Rate:  0.0065070680221620035
Quality:  0.9680292808961178
```

Final Results :

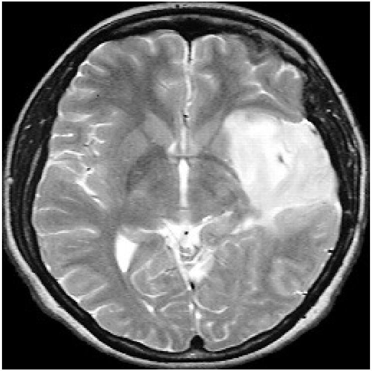

Figure: Input

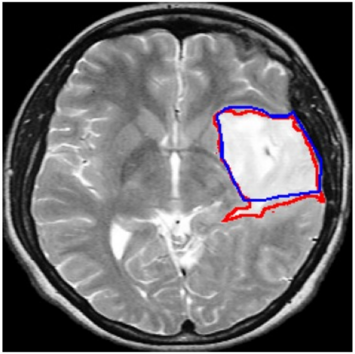

Figure: Red line: Outline given by algorithm
Blue line: Ground truth outline

```
True Positive Rate:  0.9367327667610954
False Positive Rate:  0.009037520391517128
Quality:  0.9548091846912835
```

Final Results :

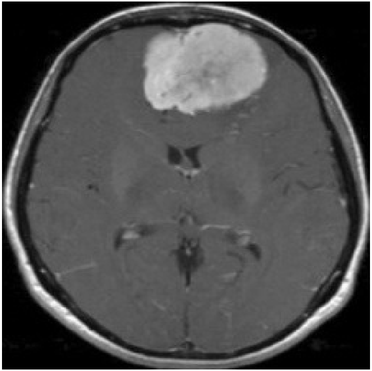

Figure: Input

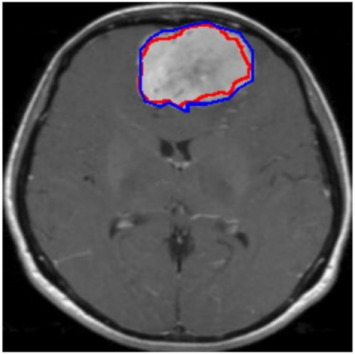

Figure: Red line: Outline given by algorithm
Blue line: Ground truth outline

```
True Positive Rate:  0.8523657870791629
False Positive Rate:  0.00019627085377821394
Quality:  0.8956067546557654
```

References

"A semi-automatic brain tumor segmentation algorithm," [Link]