

COMPUTER VISION AND IMAGING

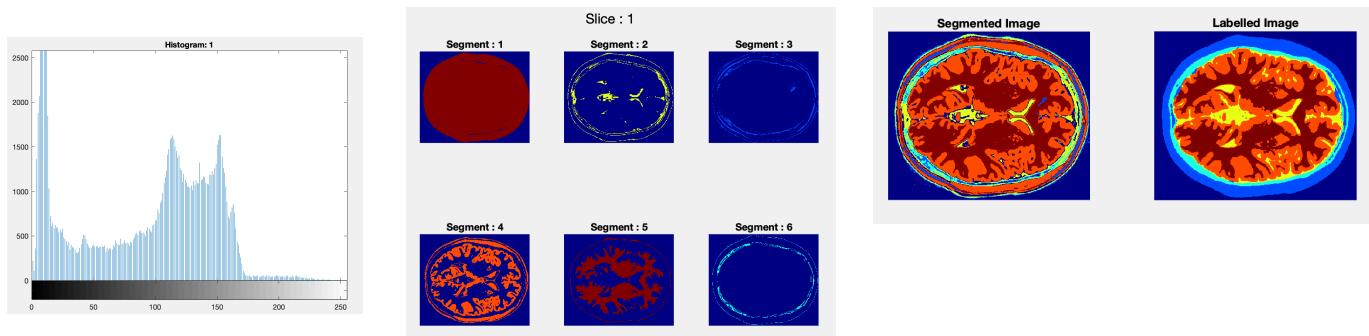
Taanya Sunil (2345404)

Aim: The aim of this assignment was to implement multiple image segmentation algorithms to ten 2D slices of the brain and carry out analysis to see which algorithm produces the best results.

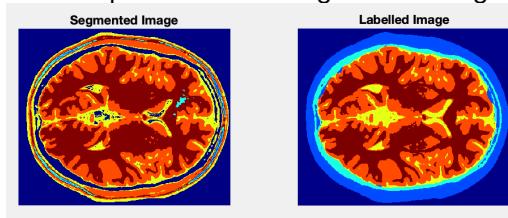
Method:

- A) **2D Brain Scans:** For the purpose of this experiment, 6 image segmentation algorithms were implemented on ten 2D brain scans. The 6 methods implemented are:

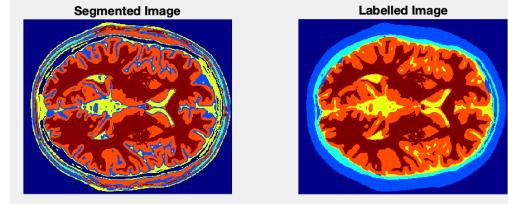
1. **Thresholding:** The task involves segmenting the brain images into 6 segments. Due to this standard thresholding cannot be applied to the image as it can work only on binary images. For this reason **imhist(I)** was used to visualise the histogram of the image in order to gain an understanding of how the pixels of the image are divided into 6 classes. Otsu thresholding was then implemented on all ten slices using these ranges of pixel values and the image was segmented into 6 classes. The segmented image of the first slice can be found below.



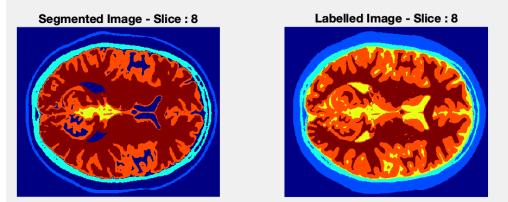
2. **Multilevel Thresholding:** Multilevel Thresholding is a process that segments a gray scale image into multiple classes. It determines several thresholds depending on the number of classes provided by the user(N) and then segments the image into multiple brightness levels. For this **grayslice(I,N)** was used which returns the image with multiple labels. The segmented image of the first slice can be found below.



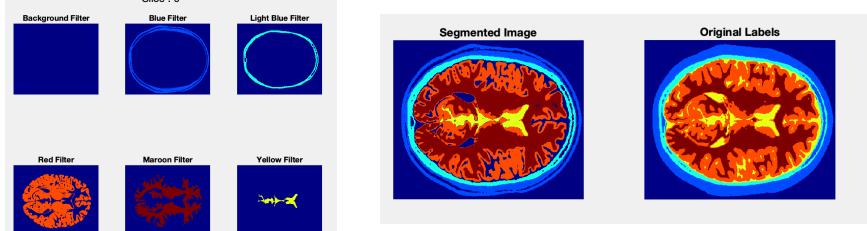
3. **KMeans:** Kmeans is a form of clustering that partitions the data points into a number of clusters in which each pixel belongs to the cluster with the nearest mean. **imsegkmeans(I,k)** was implemented on all 10 slices where k represents the number of clusters : 6. There are 2 hyper parameters that affect the performance of this algorithm. The first one is 'NumAttempts' which dictates the number of times the clustering process must be repeated (default:3) and the MaxIterations which defines the maximum number of iterations(default:100). Through this experiment, it was found that a NumAttempts value of 5 and 6 were giving the best performance. It was also found that among these two, the model reached optimum faster at a NumAttempts value of 6 as it took only 19 iterations to achieve the same performance the model with NumAttempts value of 5 took after 22 iterations. Hence the algorithm was implemented with a NumAttempts value of 6. The segmented image of the first slice can be found below.



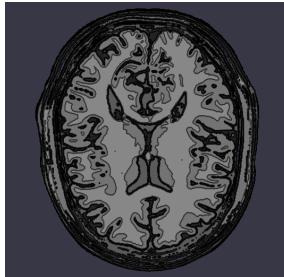
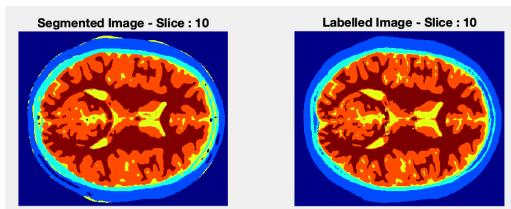
4. **Flood-Fill Technique:** This is a pixel based segmentation algorithm that segments the image by finding areas according to the difference between their pixel gray values. It is called a pixel based segmentation algorithm as it involves the selection of initial seed points from which the segments are found. For this algorithm **grayconnected** was used which finds segments of an image that have similar gray values using the aforementioned flood-fill technique. The segmented image of a slice can be found below.



5. Region Growing: This is another pixel based segmentation algorithm. It involves the selection of initial seed points which are compared to the unlabelled neighbouring pixels in the region. The difference between a pixel's intensity value and the regions mean is used as a measure of similarity. The pixel with the smallest difference measured in this way is allocated to a particular region. The process stops when the intensity difference between the region mean and the pixel intensity exceeds a certain threshold. For this algorithm we have to specify both a seed point (x,y) of a particular segment as well as a certain threshold which we use for comparison. The segmented image of one slice as well as the individual filters can be found below.



6. Combined Model: This model involved a combination of all 5 of the aforementioned models and also implemented image processing methods like smoothing using a median filter for larger regions like Air. This algorithm also involves the use of a filter called modfilt which takes a pixel and outputs a pixel that has the value of the mode in the area. For this, several filter sizes were used and a size of [5,5] was found to give the best result. The segmented image of a slice can be found below.



- B) 3D Brain Scan:** The algorithm chosen to implement image segmentation on the 3D scan is a combination of KMeans and Thresholding. They were implemented separately and then together and the results are shown below. 'modfilt' and 'medfilt3' were implemented here as well with a filter size of [5,5,5] or [1,1,1] depending on the size of the area applied to. Similar to the 2D Kmeans it was found that NumAttempts had no effect on the performance and the algorithm reached its optimum at the 35th iteration.

Evaluation : There are multiple evaluation metrics that can be implemented to test the performance of an image segmentation algorithm. While metrics such as pixel accuracy, global accuracy and meanaccuracy do work, they aren't very reliable in cases where the data is imbalanced. A better metric to use in such cases is the **Dice Coefficient**. Dice Coefficient is a similarity coefficient that takes values between 0 and 1 (1 showing greatest similarity). The Dice score not only a measure of the number of positives that we find but it also penalises the algorithm for the false positives it finds. Besides this, the **Generalised Dice score** has also been used as an evaluation metric. It is based on the Sørensen-Dice similarity and controls the weight assigned to each class by weighting classes by the inverse size of the expected region. It is especially useful when working with imbalanced datasets as it helps prevent the bigger classes from dominating the similarity score. The dice score gives the similarity score for each individual class whereas Generalised dice score gives us the dice score for the whole image.

Results:

The following is a table that outline the generalised Dice score as well as all the individual dice scores for each of the classes for all the methods used:

Algorithm	Air	Skin/Scalp(B)	Skull(C)	CSF(Y)	Gray Matter(R)	White Matter(M)	Generalised Dice Score
Thresholding	0.94591	0.18015	0.54698	0.33153	0.76414	0.84945	0.22162
KMeans	0.91038	0.34558	0	0.50614	0.81792	0.94123	0.40937
Multi-Level Thresholding	0.89206	0.067366	0	0.66391	0.7541	0.81732	0.61074
Flood-Fill Technique	0.99983	0.5543	0.86256	0.36918	0.56915	0.77539	0.92412
Region Growing	1	0.84335	0.90473	0.35006	0.74842	0.83436	0.91802
KMeans3D	0.91497	0.33837	0	0.49232	0.82208	0.95352	0.92484
Thresholding3D	0.97247	0.23882	0.69307	0.37742	0.7668	0.90475	0.93925
Combined (3D)	0.9679	0.36653	0.59137	0.46004	0.82249	0.91155	0.93981
Combined (2D)	0.9898	0.90075	0.94386	0.77581	0.9262	0.97538	0.90841

Conclusion:

From the table above, it is evident that most algorithms face an issue properly segmenting the Skin/Scalp, Skull and CSF. **Thresholding:** The poor performance of the thresholding algorithm can be attributed to the fact that retrieving the pixel values from the histogram is a cumbersome and error-prone process. From the individual similarity scores of the pixels it can be concluded that the classes that were most difficult to segment are the Skin/Scalp and the CSF. **KMeans:** Although KMeans is one of the most used algorithms for image segmentation it doesn't seem to perform as well as the other algorithms. Despite performing hyper-parameter tuning, the results don't seem very exciting. It failed to segment the skull accurately at all.

Multi-level Thresholding: Multilevel Thresholding was implemented using `grayslice` and was able to produce an average output. This too failed to segment the skull accurately. **Flood-Filled Technique and**

Region Growing: Based on the Generalised Dice Score, it can be said that the Flood-filled Technique performed the best. However, when we take a closer look at the individual classes, the same cannot be said. This is because this metric weights classes by the inverse of the class region. Although the Region Growing algorithm has performed well on all the individual classes with exception of CSF, it still has a lower generalised dice score than Flood-Fill Technique. This is because it has a slightly worse performance in CSF which is a very small region in the graph thus having greater weight. Hence, although it outperforms the flood fill technique in all other classes, it doesn't have a higher overall dice score as those regions have lower weights. A similar argument can be made in favour of the Combined Algorithm. This model has performed exceptionally well in all classes and yet has lower general dice score. Hence, it is safe to say that conclusions about which model is the best cannot be made by solely looking at the general dice score. Properly analysing the individual class dice scores is equally if not more important.

It is important to note that, the performance of the Region Growing model could possibly be improved even more if various values between 0.01 and 0.4 were entered in place of the threshold and tested to find the highest dice score. This was attempted, however the number of values to be tried was very large and hence it was computationally very expensive. Hence for this experiment, some values were found manually for the slices that provided a good result and those were implemented to get the very high dice score as shown above. This score could possibly be improved in the presence of computational power. However, accuracy-speed tradeoff is something that seems to be a problem for this model. Similarly, the performance of the combined model can also be improved in the parts that implemented region growing by testing various hyper-parameters. **KMeans3D:** This algorithm performed quite well. However it was unable to properly identify the skull. It also did show a massive improvement compared to its 2D counterpart despite carrying out the same algorithm. This could have to do with the fact that all the slices were analysed together instead of individually as in the 2D case. **Thresholding3D** was also implemented alone before combining it with KMeans3D and had better performance in some segments. Hence, they were combined and this has lead to a much better model than KMeans and Thresholding alone as in KMeans the skull was not detected which could be fixed and in thresholding, the segmentation of the scalp/skin showed improvement.

Overall, the 3D combined model has a slightly worse performance than the 2D combined model proposed. The 2D combined model consistently outperforms this model especially when it comes to identifying the skull, skin/scalp and CSF, which are the 3 regions of the brain that cause difficulty in segmentation. It can be concluded from the table that the 3D model performs just as well as the 2D model in 3 out of 6 classes excluding the skull, skin/scalp and CSF. These 3 regions were segmented using Region Growing and MultiLevel Thresholding in the 2D case and the superior performance of the 2D model can be attributed to this. It could also be concluded from the table above that pixel based methods like Region Growing and Flood-Fill techniques do outperform methods that implement Thresholding or MultiLevel Thresholding or Clustering like KMeans. This could also be why the 2D combined model performed better than the 2D one. Using blurring filters on some algorithms like Region Growing and Flood Filled Technique did lead to a worse performance. However combining multiple methods and implementing image processing techniques like blurring with the help of filters as in the CombinedMethods does lead to the best results.

SUPPLEMENTARY PAGES:

1. Thresholding:

```

1 % Displaying histograms for all the images :
2 background = zeros(1,10);
3 blue = zeros(1,10);
4 cyan = zeros(1,10);
5 yellow = zeros(1,10);
6 red = zeros(1,10);
7 maroon = zeros(1,10);
8 gen = zeros(1,10);
9 figure
10 % Displaying histogram of first slice:
11 file = imread(T1_file(1));
12 imhist(file)
13 title('Histogram: 1')
14 for m = 1:10
15     file = imread(T1_file(m));
16     mat = zeros(362,434);
17     matrix = zeros(362,434);
18     %% Background
19     for i = 1:362
20         for j = 1:434
21             if file(i,j) >= 0 && file(i,j) <= 15
22                 mat(i,j) = 1;
23                 matrix(i,j) = mat(i,j);
24             else
25                 mat(i,j) = 255;
26             end
27         end
28     end
29     map = jet;
30     figure;
31     subplot(2,3,1)
32     imshow(mat,map)
33     title('Segment : 1')
34     %% Yellow portion
35     for i = 1:362
36         for j = 1:434
37             if file(i,j) >= 30 && file(i,j) <= 55
38                 mat(i,j) = 153;
39                 matrix(i,j) = mat(i,j);
40             else
41                 mat(i,j) = 0;
42             end
43         end
44     end
45     map = jet;
46     subplot(2,3,2)
47     imshow(mat,map)
48     title('Segment : 2')
49     %% Blue portion
50     for i = 1:362
51         for j = 1:434
52             if (file(i,j) >= 15 && file(i,j) <= 21) || (file(i,j) >= 170 && file(i,j) <= 255)
53                 mat(i,j) = 51;
54                 matrix(i,j) = mat(i,j);
55

```

Move here to reveal toc

```

55
56     else
57         mat(i,j) = 0;
58     end
59 end
60 map = jet;
61 subplot(2,3,3)
62 imshow(mat,map)
63 title('Segment : 3')
64 %% Red portion
65 for i = 1:362
66     for j = 1:434
67         if file(i,j) >= 80 && file(i,j) <= 130
68             mat(i,j) = 204;
69             matrix(i,j) = mat(i,j);
70         else
71             mat(i,j) = 0;
72         end
73     end
74 end
75 map = jet;
76 subplot(2,3,4)
77 imshow(mat,map)
78 title('Segment : 4')
79 %% Maroon portion
80 for i = 1:362
81     for j = 1:434
82         if file(i,j) >= 130 && file(i,j) <= 170
83             mat(i,j) = 255;
84             matrix(i,j) = mat(i,j);
85         else
86             mat(i,j) = 0;
87         end
88     end
89 end
90 map = jet;
91 subplot(2,3,5)
92 imshow(mat,map)
93 title('Segment : 5')
94 %% Cyan portion
95 for i = 1:362
96     for j = 1:434
97         if file(i,j) >= 20 && file(i,j) <= 29
98             mat(i,j) = 102;
99             matrix(i,j) = mat(i,j);
100        else
101            mat(i,j) = 0;
102        end
103    end
104 end
105 map = jet;
106 subplot(2,3,6)
107 imshow(mat,map)
108 title('Segment : 6')

```

```

109 sgtitle(sprintf('Slice : %d',m))
110 % Labels:
111 Lab = imread(Labels_file(m));
112 for i = 1:362
113     for j = 1:434
114         if Lab(i,j) == 0
115             Lab(i,j) = 1;
116         end
117     end
118 end
119 % Displaying the segmented image and the label
120 figure;
121 subplot(1,2,1)
122 imshow(matrix,map)
123 title('Segmented Image')
124 subplot(1,2,2)
125 imshow(Lab,map)
126 title('Labelled Image')
127 sgtitle(sprintf('Slice : %d',m))
128 % Checking dice score:
129 Label_1double = double(Lab);
130 matrix = double(matrix);
131 similarity = dice(matrix, Label_1double);
132 similarity2 = generalizedDice(matrix, Label_1double);
133
134 background(m) = similarity(1);
135 blue(m) = similarity(51);
136 cyan(m) = similarity(102);
137 yellow(m) = similarity(153);
138 red(m) = similarity(204);
139 maroon(m) = similarity(255);
140 gen(m) = similarity2;
141
142 back = mean(background);
143 blues = mean(blue);
144 cyans = mean(cyan);
145 yellows = mean(yellow);
146 reds = mean(red);
147 maroons = mean(maroon);
148 gens = mean(gen);
149 disp(["Similarity value of Air (Background):" back]);
150 disp(["Similarity value of Skin/Scalp (Blue):" blues]);
151 disp(["Similarity value of Skull (Cyan):" cyans]);
152 disp(["Similarity value of CSF (Yellow):" yellows]);
153 disp(["Similarity value of Gray Matter (Red):" reds]);
154 disp(["Similarity value of White Matter (Maroon):" maroons]);
155 disp(["The Generalized Dice Score is :" gens]);

```

2. Multi-Level Thresholding

```

1 %% Multilevel thresholding:
2 background = zeros(1,10);
3 blue = zeros(1,10);
4 cyan = zeros(1,10);
5 yellow = zeros(1,10);
6 red = zeros(1,10);
7 maroon = zeros(1,10);
8 gen = zeros(1,10);
9 for m = 1:10
10     file = imread(T1_file(m));
11     X = grayslice(file,6);
12     X = uint8(X);
13     for i = 1:362
14         for j = 1:434
15             if X(i,j) == 1
16                 X(i,j) = 153;
17             elseif X(i,j) == 3
18                 X(i,j) = 255;
19             elseif X(i,j) == 2
20                 X(i,j) = 204;
21             elseif X(i,j) == 4
22                 X(i,j) = 102;
23             elseif X(i,j) == 5
24                 X(i,j) = 51;
25             elseif X(i,j) == 0
26                 X(i,j) = 1;
27             end
28         end
29     end
30     Lab = imread(Labels_file(m));
31     % Changing background values from 0 to 1 as dice takes index values from 1
32     for i = 1:362
33         for j = 1:434
34             if Lab(i,j) == 0
35                 Lab(i,j) = 1;
36             end
37         end
38     end
39     X = double(X);
40     Label_1double = double(Lab);
41     similarity = dice(X, Label_1double);
42     similarity2 = generalizedDice(X, Label_1double);
43
44     figure;
45     subplot(1,2,1)
46     imshow(X, map)
47     axis off
48     title('Segmented Image')
49     subplot(1,2,2)
50     imshow(Lab, map)
51     axis off
52     title('Labelled Image')
53     sgtitle(sprintf('Slice : %d',m))
54     impixelinfo
55
56     background(m) = similarity(1);
57     blue(m) = similarity(51);
58     cyan(m) = similarity(102);
59     yellow(m) = similarity(153);
60     red(m) = similarity(204);
61     maroon(m) = similarity(255);
62     gen(m) = similarity2;
63
64     back = mean(background);
65     blues = mean(blue);
66     cyans = mean(cyan);
67     yellows = mean(yellow);
68     reds = mean(red);
69     maroons = mean(maroon);
70     gens = mean(gen);
71     disp(["Similarity value of Air (Background):" back]);
72     disp(["Similarity value of Skin/Scalp (Blue):" blues]);
73     disp(["Similarity value of Skull (Cyan):" cyans]);
74     disp(["Similarity value of CSF (Yellow):" yellows]);
75     disp(["Similarity value of Gray Matter (Red):" reds]);
76     disp(["Similarity value of White Matter (Maroon):" maroons]);
77     disp(["The Generalized Dice Score is :" gens]);

```

3. KMeans

```

1 % Implementing Kmeans;
2 background = zeros(1,10);
3 blue = zeros(1,10);
4 cyan = zeros(1,10);
5 yellow = zeros(1,10);
6 red = zeros(1,10);
7 maroon = zeros(1,10);
8 gen = zeros(1,10);
9
10 for m = 1:10
11     file = imread(T1_file(m));
12     J = imsegkmeans(file,6,NumAttempts=6);
13 % Converting the labels to match the labels provided in order to analyse them.
14 if m == 1
15     for i = 1:362
16         for j = 1:434
17             if L(i,j) == 2; L(i,j) = 1;elseif L(i,j) == 3 ; L(i,j) = 51;elseif L(i,j) == 1 ; L(i,j) = 255;elseif L(i,j) == 4 ; L(i,j) = 153;elseif L(i,j) == 5 ; L(i,j) = 204;elseif L(i,j) == 6 ; L(i,j) = 102;
18             end
19         end
20     end
21 elseif m == 2
22     for i = 1:362
23         for j = 1:434
24             if L(i,j) == 2;L(i,j) = 204;elseif L(i,j) == 3 ; L(i,j) = 51;elseif L(i,j) == 1 ; L(i,j) = 153;elseif L(i,j) == 4 ; L(i,j) = 102;elseif L(i,j) == 5 ; L(i,j) = 255;elseif L(i,j) == 6 ; L(i,j) = 153;
25             end
26         end
27     end
28 elseif m == 3
29     for i = 1:362
30         for j = 1:434
31             if L(i,j) == 2; L(i,j) = 1;elseif L(i,j) == 3 ; L(i,j) = 51;elseif L(i,j) == 1; L(i,j) = 153;elseif L(i,j) == 4; L(i,j) = 102;elseif L(i,j) == 5 ; L(i,j) = 255;elseif L(i,j) == 6 ; L(i,j) = 204;
32             end
33         end
34     end
35 elseif m == 4
36     for i = 1:362
37         for j = 1:434
38             if L(i,j) == 2;L(i,j) = 255;elseif L(i,j) == 3;L(i,j) = 1;elseif L(i,j) == 1;L(i,j) = 153;elseif L(i,j) == 4;L(i,j) = 51;elseif L(i,j) == 5;L(i,j) = 284;elseif L(i,j) == 6;L(i,j) = 102;
39             end
40         end
41     end
42 elseif m == 5
43     for i = 1:362
44         for j = 1:434
45             if L(i,j) == 2;L(i,j) = 204; elseif L(i,j) == 3;L(i,j) = 153;elseif L(i,j) == 1;L(i,j) = 1; elseif L(i,j) == 4 ;L(i,j) = 255; elseif L(i,j) == 5 ;L(i,j) = 51; elseif L(i,j) == 6;L(i,j) = 102;
46             end
47         end
48     end
49 elseif m == 6
50     for i = 1:362
51         for j = 1:434
52             if L(i,j) == 2; L(i,j) = 51;elseif L(i,j) == 3;L(i,j) = 255; elseif L(i,j) == 1;L(i,j) = 1;elseif L(i,j) == 4 ;L(i,j) = 153;elseif L(i,j) == 5 ;L(i,j) = 284;elseif L(i,j) == 6;L(i,j) = 102;
53             end
54         end
55     end
56 elseif m == 7
57     for i = 1:362
58         for j = 1:434
59             if L(i,j) == 2; L(i,j) = 204;elseif L(i,j) == 3;L(i,j) = 255; elseif L(i,j) == 1;L(i,j) = 1;elseif L(i,j) == 4 ;L(i,j) = 51;elseif L(i,j) == 5 ;L(i,j) = 153;elseif L(i,j) == 6;L(i,j) = 102;
60             end
61         end
62     end
63 elseif m == 8
64     for i = 1:362
65         for j = 1:434
66             if L(i,j) == 2; L(i,j) = 255;elseif L(i,j) == 3;L(i,j) = 153; elseif L(i,j) == 1 ; L(i,j) = 51;elseif L(i,j) == 4 ;L(i,j) = 284;elseif L(i,j) == 5 ;L(i,j) = 1;elseif L(i,j) == 6;L(i,j) = 102;
67             end
68         end
69     end
70 elseif m == 9
71     for i = 1:362
72         for j = 1:434
73             if L(i,j) == 2; L(i,j) = 204;elseif L(i,j) == 3;L(i,j) = 1; elseif L(i,j) == 1; L(i,j) = 153;elseif L(i,j) == 4 ;L(i,j) = 51;elseif L(i,j) == 5 ;L(i,j) = 102;elseif L(i,j) == 6;L(i,j) = 255;
74             end
75         end
76     end
77 elseif m == 10
78     for i = 1:362
79         for j = 1:434
80             if L(i,j) == 2; L(i,j) = 204;elseif L(i,j) == 3;L(i,j) = 51; elseif L(i,j) == 1; L(i,j) = 153;elseif L(i,j) == 4 ;L(i,j) = 51;elseif L(i,j) == 5 ;L(i,j) = 102;elseif L(i,j) == 6;L(i,j) = 255;
81             end
82         end
83     end
84 end
85
86 Lab = imread(Label1_file(m));
87 % Changing Labels background values from 0 to 1 as dice takes index values from 1
88 for i = 1:362
89     for j = 1:434
90         if Lab(i,j) == 0;
91             Lab(i,j) = 1;
92         end
93     end
94 end
95 L = modefilt(L, [5,5]);
96 % Converting to double in order to analyse
97 Label_1ddouble = double(Lab);
98 similarity = dice1, Label_1ddouble;
99 similarity2 = generalizedDice1, Label_1ddouble;
100
101 figure;
102 map = jet;
103 subplot(1,2,1)
104 subplot(1,2,2)
105 imshow(map)
106 imixelinfo
107 axis off
108 title("Segmented Image")
109
110 subplot(1,2,2)
111 imshow(Lab, map)
112 imixelinfo
113 title("Labelled Image")
114 spltile(sprintf("Slice : %d",m))
115
116 background = similarity(1);
117 blues = similarity(51);
118 cyan = similarity(102);
119 gen0 = similarity2;
120
121 meanbackground;
122 blues = mean(blues);
123 cyan = mean(cyan);
124 yellows = mean(yellow);
125 reds = mean(red);
126 meanmaroon;
127 gens = mean(gen);
128
129 disp("Similarity value of Air (Background):" back);
130 disp("Similarity value of Skin/Scalp (Blues):" blues);
131 disp("Similarity value of CSF (Cyan):" cyan);
132 disp("Similarity value of CSF (Yellow):" yellow);
133 disp("Similarity value of Gray Matter (Red):" reds);
134 disp("Similarity value of White Matter (Maroon):" maroons);
135
136 disp("The Generalized Dice Score is :" gens);
137
138

```

4. Flood-fill Technique

```

1 %% Implementing Flood-Fill Technique:
2 background = zeros(1,10);
3 blue = zeros(1,10);
4 cyan = zeros(1,10);
5 yellow = zeros(1,10);
6 red = zeros(1,10);
7 maroon = zeros(1,10);
8 gen = zeros(1,10);
9 for m = 1:10
10     img = imread(T1_file(m)); % BACKGROUND
11     Mat = zeros(362,434);
12
13 J = grayconnected(img,34,34); % BACKGROUND
14 J = J*1;
15 J = uint8(J);
16 for i = 1:362
17     for j = 1:434
18         if J(i,j) == 1
19             Mat(i,j) = J(i,j);
20         end
21     end
22 end
23 K = grayconnected(img,180,22); % BLUE
24 K = K*51;
25 K = uint8(K);
26 for i = 1:362
27     for j = 1:434
28         if K(i,j) == 51
29             Mat(i,j) = K(i,j);
30         end
31     end
32 end
33 Q = grayconnected(img,96,76); % LIGHT BLUE
34 Q = Q*102;
35 Q = uint8(Q);
36 for i = 1:362
37     for j = 1:434
38         if Q(i,j) == 102
39             Mat(i,j) = Q(i,j);
40         end
41     end
42 end
43 N = grayconnected(img,144,64); % RED
44 N = N*204;
45 N = uint8(N);
46 for i = 1:362
47     for j = 1:434
48         if N(i,j) == 204
49             Mat(i,j) = N(i,j);
50         end
51     end
52 end
53 X = grayconnected(img,234,222); % MAROON
54 X = X*255;
55 X = uint8(X);
56 for i = 1:362
57     for j = 1:434
58         if X(i,j) == 255
59             Mat(i,j) = X(i,j);
60         end
61     end

```

```

62 end
63
64 I = grayconnected(img,172,178); % YELLOW
65 I = I*153;
66 I = uint8(I);
67 for i = 1:362
68     for j = 1:434
69         if I(i,j) == 153
70             Mat(i,j) = I(i,j);
71         end
72     end
73 end
74
75 % Changing background values from 0 to 1 as dice takes index values from 1
76 Lab = imread(Labels_file(m));
77 for i = 1:362
78     for j = 1:434
79         if Lab(i,j) == 0
80             Lab(i,j) = J(i,j);
81         end
82     end
83 end
84
85 % Converting to double in order to analyse
86 Label_1double = double(Lab);
87 similarity = dice(Mat, Label_1double);
88 similarity2 = generalizedDice(Mat, Label_1double);
89
90 map = jet;
91 figure
92 subplot(1,2,1)
93 imshow(Mat, map);
94 impixelinfo;
95 title(sprintf('Segmented Image - Slice : %d',m))
96 %title(['The Generalized Dice Score is = ' num2str(similarity2)]);
97
98 subplot(1,2,2)
99 imshow(Lab, map)
100 impixelinfo;
101 title(sprintf('Labelled Image - Slice : %d',m))
102
103 background(m) = similarity(1);
104 blue(m) = similarity(51);
105 cyan(m) = similarity(102);
106 yellow(m) = similarity(153);
107 red(m) = similarity(204);
108 maroon(m) = similarity(255);
109 gen(m) = similarity2;
110 end
111 back = mean(background);
112 blues = mean(blue);
113 cyans = mean(cyan);
114 yellows = mean(yellow);
115 reds = mean(red);
116 maroons = mean(maroon);
117 gens = mean(gen);
118 disp(["Similarity value of Air (Background):" back]);
119 disp(["Similarity value of Skin/Scalp (Blue):" blues]);
120 disp(["Similarity value of Skull (Cyan):" cyans]);
121 disp(["Similarity value of CSF (Yellow):" yellows]);
122 disp(["Similarity value of Gray Matter (Red):" reds]);

```

```

title('Light Blue Filter')
for i = 1:362
    for j = 1:434
        if X(i,j) == 102
            Mat(i,j) = X(i,j);
        end
    end
end

% Doing it for the next section: Red
x=262; y=282;
if m == 1; val = 0.11; elseif m == 10; val = 0.2; else val = 0.11; end
N = regiongrowing(I,x,y,val);
N = uint8(N);
%K = imadjust(I,[0 1]);
subplot(2,3,4)
imshow(N);
colormap('jet')
title('Red Filter')

for i = 1:362
    for j = 1:434
        if N(i,j) == 204
            Mat(i,j) = N(i,j);
        end
    end
end

% Doing it for the next section: Maroon
x=148; y=264;
K = regiongrowing(I,x,y,0.09);
K = K*255;
K = uint8(K);
%K = imadjust(I,[0 1]);
subplot(2,3,5)
imshow(K);
impixelinfo;
colormap('jet')
title('Maroon Filter')

for i = 1:362
    for j = 1:434
        if K(i,j) == 255
            Mat(i,j) = K(i,j);
        end
    end
end

% Doing it for the next section: Yellow
x=178; y=178;
if m == 1; val = 0.13; elseif m == 2; val = 0.12; elseif m == 3 ; val = 0.12; elseif m == 4; val = 0.14;elseif m == 5; val = 0.14; elseif m == 6; val = 0.14;elseif m == 7; val = 0.13;elseif m == 8; val = 0.14;elseif m == 9; val = 0.14; elseif m ==10; val = 0.09; M = regiongrowing(I,x,y,val);
M = M*153;
M = uint8(M);
subplot(2,3,6)
imshow(M);
colormap('jet')
title('Yellow Filter')

```

5. Region Growing

```

1 background = zeros(1,10);
2 blu = zeros(1,10);
3 cyan = zeros(1,10);
4 yellow = zeros(1,10);
5 red = zeros(1,10);
6 maroon = zeros(1,10);
7 gen = zeros(1,10);
8 for m = 1:10
9 % First doing it only for the background
10 I = im2double(imread(T1_file(m)));
11 x=50; y=50;
12 J = regiongrowing(I,x,y,0.1);
13 J = J*1;
14 J = uint8(J);
15 figure;
16 subplot(2,3,1)
17 imshow(J);
18 impixelinfo;
19 colormap('jet')
20 title('Background Filter')
21
22 Mat = zeros(362,434);
23
24 % Changing background values from 0 to 1 as dice takes index values from 1
25 for i = 1:362
26     for j = 1:434
27         if J(i,j) == 1
28             Mat(i,j) = J(i,j);
29         end
30     end
31 end
32
33 % Doing it for the next section: Blue
34 x=252; y=36;
35 %if m==1; val = 0.3; elseif m ==2; val = 0.3; end
36 D = regiongrowing(I,x,y,0.2);
37 D = D*51;
38 D = uint8(D);
39 %K = imadjust(I,[0 1]);
40 subplot(2,3,2)
41 imshow(D);
42 colormap('jet')
43 title('Blue Filter')
44
45 for i = 1:362
46     for j = 1:434
47         if D(i,j) == 51
48             Mat(i,j) = D(i,j);
49         end
50     end
51 end
52
53 % Doing it for the next section: Light Blue
54 x=74; y=94;
55 X = regiongrowing(I,x,y,0.09);
56 X = X*102;
57 X = uint8(X);
58 %K = imadjust(I,[0 1]);
59 subplot(2,3,3)
60 imshow(X);
61 colormap('jet')

```

```

123 sgttitle(sprintf('Slice : %d',m))
124
125 for i = 1:362
126     for j = 1:434
127         if M(i,j) == 153
128             Mat(i,j) = M(i,j);
129         end
130     end
131 end
132
133 %% Computing Generalized Dice Score:
134 % Converting to double in order to analyse
135 Lab = imread(Labels_file(m));
136 for i = 1:362
137     for j = 1:434
138         if Lab(i,j) == 0
139             Lab(i,j) = J(i,j);
140         end
141     end
142 end
143 Label_1double = double(Lab);
144 similarity = dice(Mat, Label_1double);
145 similarity2 = generalizedDice(Mat, Label_1double);
146
147 map = jet;
148 figure
149 subplot(1,2,1)
150 imshow(Mat, map);
151 impixelinfo;
152 title('Segmented Image')
153 %title(['The Generalized Dice Score is = ' num2str(similarity)])
154
155 % Looking at the Labels
156 subplot(1,2,2)
157 imshow(Lab, map);
158 impixelinfo;
159 title('Original Labels')
160 sgttitle(sprintf('Slice : %d',m))
161
162 background(m) = similarity(1);
163 blue(m) = similarity(51);
164 cyan(m) = similarity(102);
165 yellow(m) = similarity(153);
166 red(m) = similarity(204);
167 maroon(m) = similarity(255);
168 gen(m) = similarity2;
169 end
170 back = mean(background);
171 blues = mean(blue);
172 cyans = mean(cyan);
173 yellows = mean(yellow);
174 reds = mean(red);
175 maroons = mean(maroon);
176 gens = mean(gen);
177 disp(["Similarity value of Air (Background):" back]);
178 disp(["Similarity value of Skin/Scalp (Blue):" blues]);
179 disp(["Similarity value of Skull (Cyan):" cyans]);
180 disp(["Similarity value of CSF (Yellow):" yellows]);
181 disp(["Similarity value of Gray Matter (Red):" reds]);
182 disp(["Similarity value of White Matter (Maroon):" maroons]);
183 disp(["The Generalized Dice Score is :" gens]);

```

```

1 function J=regiongrowing(I,x,y,reg_maxdist)
2 if(exist('reg_maxdist','var')==0), reg_maxdist=0.2; end
3 if(exist('y','var')==0), figure, imshow(I,[]); [y,x]=getpts; y=round(y(1)); x=round(x(1)); end
4
5 J = zeros(size(I));
6 Isizes = size(I);
7
8 reg_mean = I(x,y); % The mean of the segmented region
9 reg_size = 1;
10
11 neg_free = 10000; neg_pos=0;
12 neg_list = zeros(neg_free,3);
13
14 pixdist=0;
15 neigb=[-1 0; 1 0; 0 -1;0 1];
16

```

```

17 while(pixdist<reg_maxdist&&reg_size<numel(I))
18
19 for j=1:4
20     xn = x +neigb(j,1); yn = y +neigb(j,2);
21     ins=(xn>=1)&&(yn>=1)&&(xn<=Isizes(1))&&(yn<=Isizes(2));
22     if(ins&&(J(xn,yn)==0))
23         neg_pos = neg_pos+1;
24         neg_list(neg_pos,:) = [xn yn I(xn,yn)]; J(xn,yn)=1;
25     end
26
27 if(neg_pos+10>neg_free), neg_free=neg_free+10000; neg_list((neg_pos+1):%
28 dist = abs(neg_list(neg_pos,3)-reg_mean);
29 [pixdist, index] = min(dist);
30 J(x,y)=2; reg_size=reg_size+1;
31 reg_mean = (reg_mean*reg_size + neg_list(index,3))/(reg_size+1);
32 x = neg_list(index,1); y = neg_list(index,2);
33 neg_list(index,:)=neg_list(neg_pos,:); neg_pos=neg_pos-1;
34
35 J=J-1;

```

6. Combined 2D

```

1 %% COMBINED 2D MODEL:
2 background = zeros(1,10);
3 blue = zeros(1,10);
4 cyan = zeros(1,10);
5 yellow = zeros(1,10);
6 red = zeros(1,10);
7 maroon = zeros(1,10);
8 gen = zeros(1,10);
9 mat2 = zeros(362,434);
10 mat4 = zeros(362,434);
11 for m = 1:10
12 %img = imread(T1_file(m));
13 file = imread(T1_file(m));
14 filedoub = im2double(file);
15 file2 = medfilt2(file,[9 9]);
16 file2doub = im2double(file2);
17 Mat = zeros(362,434);
18
19 %% BACKGROUND: AIR
20 x=50; y=50;
21 J = regiongrowing(filedoub,x,y,0.1);
22 J = J>1;
23 J = uint8(J);
24 Mat = zeros(362,434);
25
26 % Changing background values from 0 to 1 as dice takes index values from 1
27 for i = 1:362
28     for j = 1:434
29         if J(i,j) == 1
30             Mat(i,j) = J(i,j);
31         end
32     end
33 end
34
35 %% CYAN
36 x=74; y=94;
37 X = regiongrowing(filedoub,x,y,0.09);
38 X = X>102;
39 X = uint8(X);
40 for i = 1:362
41     for j = 1:434
42         if X(i,j) == 102
43             Mat(i,j) = X(i,j);
44         end
45     end
46 end
47
48 %% RED
49 for i = 1:362
50     for j = 1:434
51         if file(i,j) >= 115 && file(i,j) <= 120
52             mat4(i,j) = 204;
53         else
54             mat4(i,j) = 0;
55         end
56     end
57 end
58 % Doing it for the next section: Red
59 x=194; y=201;
60 if m == 10; val = 0.2; else val = 0.09; end
61 N = N+204;
62 N = uint8(N);
63 for i = 1:362
64     for j = 1:434
65         if N(i,j) == 204
66             mat4(i,j) = N(i,j);
67             Mat(i,j) = mat4(i,j);
68         end
69     end
70 end
71
72 %% MAROON
73 [L,~] = imsegkmeans(file,6,NumAttempts=6);
74 % Correcting the labels to match the labels provided in order to analyse them
75 if m == 1
76     for i = 1:362
77         for j = 1:434
78             if L(i,j) == 1; L(i,j) = 255; Mat(i,j) = L(i,j);
79         end
80     end
81 end
82 elseif m == 2
83     for i = 1:362
84         for j = 1:434
85             if L(i,j) == 5; L(i,j) = 255;Mat(i,j) = L(i,j);
86         end
87     end
88 disp(['The Generalized Dice Score is : ' num2str(similarity2)]);
89
90 elseif m == 3
91     for i = 1:362
92         for j = 1:434
93             if L(i,j) == 5; L(i,j) = 255;Mat(i,j) = L(i,j);
94         end
95     end
96 end
97 elseif m == 4
98     for i = 1:362
99         for j = 1:434
100            if L(i,j) == 2;L(i,j) = 255;Mat(i,j) = L(i,j);
101        end
102    end
103 end
104 elseif m == 5
105     for i = 1:362
106         for j = 1:434
107             if L(i,j) == 4 ;L(i,j) = 255;Mat(i,j) = L(i,j);
108         end
109     end
110 end
111 elseif m == 6
112     for i = 1:362
113         for j = 1:434
114             if L(i,j) == 3;L(i,j) = 255;Mat(i,j) = L(i,j);
115         end
116     end
117 end
118 elseif m == 7
119     for i = 1:362
120         for j = 1:434
121             if L(i,j) == 3;L(i,j) = 255;Mat(i,j) = L(i,j);
122         end
123     end

```

```

123
124
125 elseif m == 8
126     for i = 1:362
127         for j = 1:434
128             if L(i,j) == 2; L(i,j) = 255;Mat(i,j) = L(i,j);
129         end
130     end
131 end
132 elseif m == 9
133     for i = 1:362
134         for j = 1:434
135             if L(i,j) == 6;L(i,j) = 255;Mat(i,j) = L(i,j);
136         end
137     end
138 end
139 elseif m == 10
140     for i = 1:362
141         for j = 1:434
142             if L(i,j) == 4 ;L(i,j) = 255;Mat(i,j) = L(i,j);
143         end
144     end
145 end
146 end
147
148 %% YELLOW
149 X = grayslice(file,6);
150 X = uint8(X);
151 for i = 1:362
152     for j = 1:434
153         if X(i,j) == 1
154             X(i,j) = 153;
155             Mat(i,j) = X(i,j);
156         end
157     end
158 end
159
160 %% BLUE
161 for i = 1:362
162     for j = 1:434
163         if file2(i,j) >= 170 && file2(i,j) <= 255
164             mat2(i,j) = 51;
165         else
166             mat2(i,j) = 0;
167         end
168     end
169 end
170 x=73; y=68;
171 D = regiongrowing(file2doub,x,y,0.18);
172 D = D>51;
173 D = uint8(D);
174 for i = 1:362
175     for j = 1:434
176         if D(i,j) == 51
177             mat2(i,j) = D(i,j);
178             Mat(i,j) = mat2(i,j);
179         end
180     end
181 end
182
183 Mat = modefilt(Mat,[5 5]);
184
185 % Changing background values from 0 to 1 as dice takes index values from 1
186 Lab = imread(Labels_file(m));
187 for i = 1:362
188     for j = 1:434
189         if Lab(i,j) == 0
190             Lab(i,j) = J(i,j);
191         end
192     end
193 end
194
195 % Converting to double in order to analyse
196 Label_1double = double(Lab);
197 similarity = dice(Mat, Label_1double);
198 similarity2 = generalizedDice(Mat, Label_1double);
199
200 map = jet;
201 figure
202 subplot(1,2,1)
203 imshow(Mat, map);
204 impixelinfo;
205 title(sprintf('Segmented Image - Slice : %d',m))
206 %title(['The Generalized Dice Score is = ' num2str(similarity2)]);
207
208 subplot(1,2,2)
209 imshow(Lab, map)
210 impixelinfo;
211 title(sprintf('Labelled Image - Slice : %d',m))
212
213 background(m) = similarity(1);
214 blue(m) = similarity(51);
215 cyan(m) = similarity(102);
216 yellow(m) = similarity(153);
217 red(m) = similarity(204);
218 maroon(m) = similarity(255);
219 gen(m) = similarity2;
220
221 back = mean(background);
222 blues = mean(blue);
223 cyaans = mean(cyan);
224 yellows = mean(yellow);
225 reds = mean(red);
226 maroons = mean(maroon);
227 gens = mean(gen);
228 disp(['Similarity value of Air (Background):' back]);
229 disp(['Similarity value of Skin/Scalp (Blue):' blues]);
230 disp(['Similarity value of Skull (Cyan):' cyan]);
231 disp(['Similarity value of CSF (Yellow):' yellows]);
232 disp(['Similarity value of Gray Matter (Red):' reds]);
233 disp(['Similarity value of White Matter (Maroon):' maroons]);
234 disp(['The Generalized Dice Score is :' gens]);

```

7. KMeans 3D

```

1    volshow(T1)
2    % Segmenting the volume into 6 clusters
3    L = imsegkmeans3(T1,6, MaxIterations = 100);
4    for i = 1:362
5        for j = 1:434
6            for m = 1:10
7                if L(i,j,m) == 2
8                    L(i,j,m) = 255;
9                elseif L(i,j,m) == 3
10                   L(i,j,m) = 1;
11                elseif L(i,j,m) == 153;
12                   L(i,j,m) = 153;
13                elseif L(i,j,m) == 4
14                   L(i,j,m) = 204;
15                elseif L(i,j,m) == 5
16                   L(i,j,m) = 51;
17                elseif L(i,j,m) == 6
18                   L(i,j,m) = 102;
19                end
20            end
21        end
22    end
23    figure
24    L = modefilt(L, [5,5,5]);
25    volshow(L);
26    % Changing the label value to
27    labels = uint8(255*mat2gray(label(:,:,::)));
28    for i = 1:362
29        for j = 1:434
30            for m = 1:10
31                if labels(i,j,m) == 0
32                    labels(i,j,m) = 1;
33                end
34            end
35        end
36    end
37    for m = 1:10
38        figure
39        subplot(1,2,1)
40        imshow(labels(:,:,m));
41        title('Segmented Image')
42        subplot(1,2,2)
43        map = jet;
44        imshow(labels(:,:,m), map)
45        title('Labeled Image')
46        sgttitle(sprintf('Slice : %d',m))
47        end
48    end
49    % Calculating the overall dice value
50    Ldouble = double(L);
51    Label_1double = double(labels);
52    similarity = dice(Ldouble, Label_1double);
53    similarity2 = generalizedDice(Ldouble, Label_1double);
54
55    disp(["Similarity value of Air (Background):" similarity(1)]);
56    disp(["Similarity value of Skin/Scalp (Blue):" similarity(51)]);
57    disp(["Similarity value of Skull (Cyan):" similarity(102)]);
58    disp(["Similarity value of CSF (Yellow):" similarity(153)]);
59    disp(["Similarity value of Gray Matter (Red):" similarity(204)]);
60    disp(["Similarity value of White Matter (Maroon):" similarity(255)]);
61

```

9. Combined 3D

```

1    figure
2    volshow(T1)
3    conv = uint8(255*mat2gray(T1));
4    file = medfilt3(conv, [5,5,5]);
5    mat1 = zeros(362,434,10);mat2 = zeros(362,434,10);mat3 = zeros(362,434,10);
6    mat4 = zeros(362,434,10);mat5 = zeros(362,434,10);mat6 = zeros(362,434,10);
7    matrix = zeros(362,434,10);
8    figure
9    imhist(conv)
10
11    %% Background
12    for i = 1:362
13        for j = 1:434
14            for m = 1:10
15                if file(i,j,m) >= 0 && file(i,j,m) <= 15
16                    mat1(i,j,m) = 1;
17                    matrix(i,j,m) = mat1(i,j,m);
18                else
19                    mat1(i,j,m) = 0;
20                end
21            end
22        end
23    end
24    %% Blue
25    L = imsegkmeans3(T1,6, MaxIterations = 100);
26    for i = 1:362
27        for j = 1:434
28            for m = 1:10
29                if L(i,j,m) == 5
30                    L(i,j,m) = 51;
31                    matrix(i,j,m) = L(i,j,m);
32                end
33            end
34        end
35    end
36
37    %% Cyan portion
38    for i = 1:362
39        for j = 1:434
40            for m = 1:10
41                if file(i,j,m) >= 20 && file(i,j,m) <= 30
42                    mat6(i,j,m) = 102;
43                    matrix(i,j,m) = mat6(i,j,m);
44                else
45                    mat6(i,j,m) = 0;
46                end
47            end
48        end
49    end
50
51    %% Yellow
52    L = imsegkmeans3(T1,6, MaxIterations = 100);
53    for i = 1:362
54        for j = 1:434
55            for m = 1:10
56                if L(i,j,m) == 1
57                    L(i,j,m) = 153;
58                    matrix(i,j,m) = L(i,j,m);
59                end
60            end
61        end

```

8. Thresholding3D

```

1    %% Thresholding3D
2    figure
3    volshow(T1)
4    conv = uint8(255*mat2gray(T1));
5    file = medfilt3(conv, [5,5,5]);
6    file2 = medfilt3(file, [1,1,1]);
7    mat1 = zeros(362,434,10);mat2 = zeros(362,434,10);mat3 = zeros(362,434,10);
8    mat4 = zeros(362,434,10);mat5 = zeros(362,434,10);mat6 = zeros(362,434,10);
9    matrix = zeros(362,434,10);
10   figure
11   imhist(conv)
12
13   %% Background
14   for i = 1:362
15       for j = 1:434
16           for m = 1:10
17               if file(i,j,m) >= 0 && file(i,j,m) <= 15
18                   mat1(i,j,m) = 1;
19                   matrix(i,j,m) = mat1(i,j,m);
20               else
21                   mat1(i,j,m) = 0;
22               end
23           end
24       end
25   end
26
27   %% Yellow portion
28   for i = 1:362
29       for j = 1:434
30           for m = 1:10
31               if conv(i,j,m) >= 30 && conv(i,j,m) <= 55
32                   mat3(i,j,m) = 153;
33                   matrix(i,j,m) = mat3(i,j,m);
34               else
35                   mat3(i,j,m) = 0;
36               end
37           end
38       end
39   end
40
41   %% Red portion
42   for i = 1:362
43       for j = 1:434
44           for m = 1:10
45               if conv(i,j,m) >= 80 && conv(i,j,m) <= 130
46                   mat4(i,j,m) = 204;
47                   matrix(i,j,m) = mat4(i,j,m);
48               else
49                   mat4(i,j,m) = 0;
50               end
51           end
52   end
53
54   %% Maroon portion
55   for i = 1:362
56       for j = 1:434
57           for m = 1:10
58               if conv(i,j,m) >= 130 && conv(i,j,m) <= 170
59                   mat5(i,j,m) = 255;
60                   matrix(i,j,m) = mat5(i,j,m);
61               else
62                   mat5(i,j,m) = 0;
63               end
64           end
65       end
66   end
67
68   %% Cyan portion
69   for i = 1:362
70       for j = 1:434
71           for m = 1:10
72               if file2(i,j,m) >= 20 && file2(i,j,m) <= 29
73                   mat6(i,j,m) = 102;
74                   matrix(i,j,m) = mat6(i,j,m);
75               else
76                   mat6(i,j,m) = 0;
77               end
78           end
79       end
80   end
81
82   %% Blue
83   for i = 1:362
84       for j = 1:434
85           for m = 1:10
86               if conv(i,j,m) >= 170 && conv(i,j,m) <= 255
87                   mat2(i,j,m) = 51;
88                   matrix(i,j,m) = mat2(i,j,m);
89               else
90                   mat2(i,j,m) = 0;
91               end
92           end
93       end
94   end
95
96   %% Calculating the overall dice value
97   Ldouble = double(matrix);
98   Label_1double = double(labels);
99   similarity = dice(Ldouble, Label_1double);
100  similarity2 = generalizedDice(Ldouble, Label_1double);
101
102  disp(["Similarity value of Air (Background):" similarity(1)]);
103  disp(["Similarity value of Skin/Scalp (Blue):" similarity(51)]);
104  disp(["Similarity value of Skull (Cyan):" similarity(102)]);
105  disp(["Similarity value of CSF (Yellow):" similarity(153)]);
106  disp(["Similarity value of Gray Matter (Red):" similarity(204)]);
107  disp(["Similarity value of White Matter (Maroon):" similarity(255)]);
108
109  disp("The Generalized Dice Score is :" num2str(similarity2));

```

```

62      %% Maroon portion
63      for i = 1:362
64          for j = 1:434
65              for m = 1:10
66                  if L(i,j,m) == 102;
67                      matrix(i,j,m) = mat6(i,j,m);
68                  else
69                      matrix(i,j,m) = 0;
70                  end
71              end
72          end
73      end
74
75      %% Cyan portion
76      for i = 1:362
77          for j = 1:434
78              for m = 1:10
79                  if file2(i,j,m) >= 20 && file2(i,j,m) <= 29
80                      mat6(i,j,m) = 102;
81                      matrix(i,j,m) = mat6(i,j,m);
82                  else
83                      mat6(i,j,m) = 0;
84                  end
85              end
86          end
87      end
88
89      %% Blue
90      for i = 1:362
91          for j = 1:434
92              for m = 1:10
93                  if conv(i,j,m) >= 170 && conv(i,j,m) <= 255
94                      mat2(i,j,m) = 51;
95                      matrix(i,j,m) = mat2(i,j,m);
96                  else
97                      mat2(i,j,m) = 0;
98                  end
99              end
100         end
101     end
102
103     %% Calculating the overall dice value
104     Ldouble = double(matrix);
105     Label_1double = double(labels);
106     similarity = dice(Ldouble, Label_1double);
107     similarity2 = generalizedDice(Ldouble, Label_1double);
108
109     disp(["Similarity value of Air (Background):" similarity(1)]);
110     disp(["Similarity value of Skin/Scalp (Blue):" similarity(51)]);
111     disp(["Similarity value of Skull (Cyan):" similarity(102)]);
112     disp(["Similarity value of CSF (Yellow):" similarity(153)]);
113     disp(["Similarity value of Gray Matter (Red):" similarity(204)]);
114     disp(["Similarity value of White Matter (Maroon):" similarity(255)]);
115
116     disp("The Generalized Dice Score is :" num2str(similarity2));

```