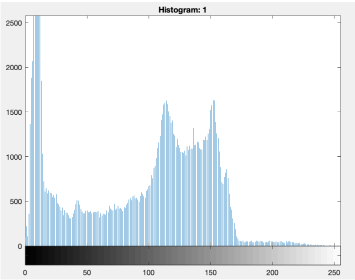
COMPUTER VISION AND IMAGING
Taanya Sunil (2345404)

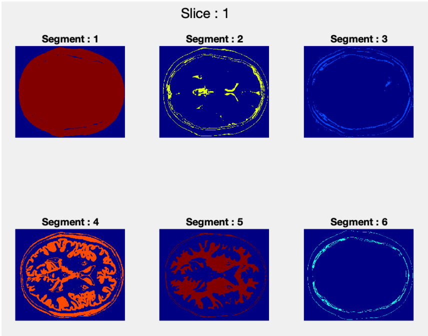
Aim : The aim of this assignment was to implement multiple image segmentation algorithms to ten 2D slices of the brain and carry out analysis to see which algorithm produces the best results.

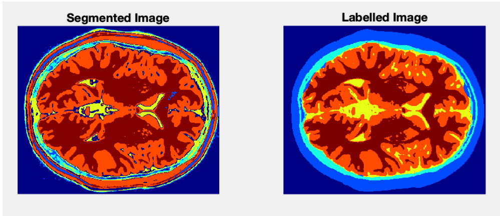
Method:

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:

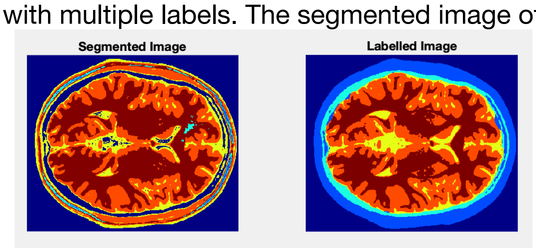
1. Thresholding: The task involves segmenting the brain images into 6 segments. Due to this standard thresholding cannot be applied to the image as it can work only on binary images. For this reason **imhist(I)** was used to visualise the histogram of the image in order to gain an understanding of how 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.



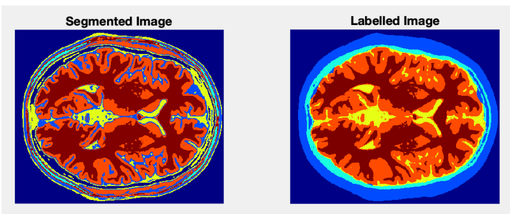




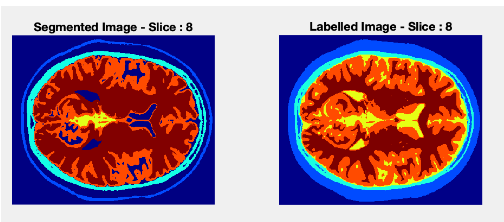
2. Multilevel Thresholding: Multilevel Thresholding is a process that segments a gray scale image into multiple classes. It determines several thresholds depending on the number of classes provided by the user(N) and then segments the image into multiple brightness levels. For this **grayslice(I,N)**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



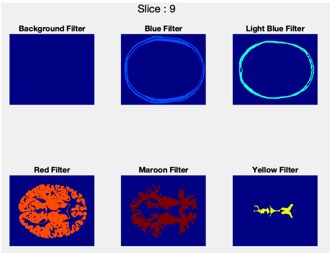
3. KMeans: Kmeans is a form of clustering that partitions the data points into a number of clusters in which each pixel belongs to the cluster with the nearest mean. **imsegkmeans(I,k)** was implemented on all 10 slices where k represents the number of clusters : 6. There are 2 hyper parameters that 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.

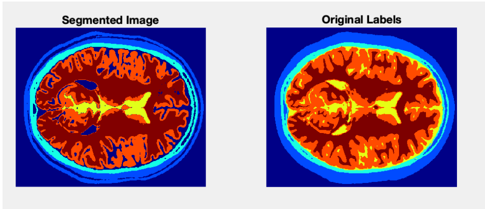


4. Flood-Fill Technique: This is a pixel based segmentation algorithm that segments the image by finding areas according to the difference between their pixel gray values. It is called a pixel based segmentation algorithm as it involves the selection of initial seed points from which the segments are 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.

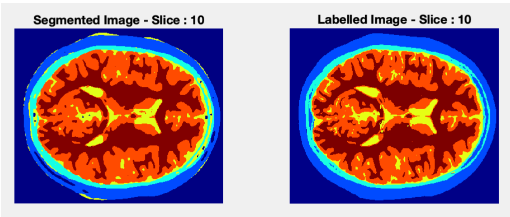


5. Region Growing: This is another pixel based segmentation algorithm. It involves the selection of initial seed points which are compared to the unlabelled neighbouring pixels in the region. The difference between a pixel's intensity value and the regions mean is used as a measure of similarity. The 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.

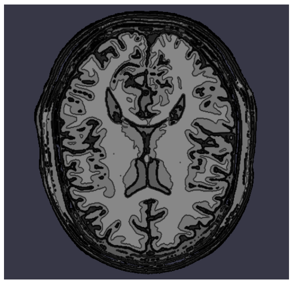




6. Combined Model: This model involved a combination of all 5 of the aforementioned models and also implemented image processing methods like smoothing using a median filter for larger regions like Air. This algorithm also involves the use of a filter called modfilt which takes a pixel and outputs a 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:

```matlab
% Displaying histograms for all the images :
background = zeros(1,10);
blue = zeros(1,10);
cyan = zeros(1,10);
yellow = zeros(1,10);
red = zeros(1,10);
maroon = zeros(1,10);
gen = zeros(1,10);
figure
% Displaying histogram of first slice:
file = imread(T1_file(1));
imhist(file)
title('Histogram: 1')
for m = 1:10
file = imread(T1_file(m));
mat = zeros(362,434);
matrix = zeros(362,434);
%% Background
for i = 1:362
    for j = 1:434
        if file(i,j) >= 0 && file(i,j) <= 15
        mat(i,j) = 1;
        matrix(i,j) = mat(i,j);
        else
        mat(i,j) = 255;
        end
    end
end
map = jet;
figure;
subplot(2,3,1)
imshow(mat,map)
title('Segment : 1')
%% Yellow portion
for i = 1:362
    for j = 1:434
        if file(i,j) >= 30 && file(i,j) <= 55
        mat(i,j) = 153;
        matrix(i,j) = mat(i,j);
        else
        mat(i,j) = 0;
        end
    end
end
map = jet;
subplot(2,3,2)
imshow(mat,map)
title('Segment : 2')
%% Blue portion
for i = 1:362
    for j = 1:434
        if (file(i,j) >= 15 && file(i,j) <= 21) || (file(i,j) >= 170 && file(i,j) <= 255)
        mat(i,j) = 51;
        matrix(i,j) = mat(i,j);
        else
        mat(i,j) = 0;
        end
    end
end
map = jet;
subplot(2,3,3)
imshow(mat,map)
title('Segment : 3')
%% Red portion
for i = 1:362
    for j = 1:434
        if file(i,j) >= 80 && file(i,j) <= 130
        mat(i,j) = 204;
        matrix(i,j) = mat(i,j);
        else
        mat(i,j) = 0;
        end
    end
end
map = jet;
subplot(2,3,4)
imshow(mat,map)
title('Segment : 4')
%% Maroon portion
for i = 1:362
    for j = 1:434
        if file(i,j) >= 130 && file(i,j) <= 170
        mat(i,j) = 255;
        matrix(i,j) = mat(i,j);
        else
        mat(i,j) = 0;
        end
    end
end
map = jet;
subplot(2,3,5)
imshow(mat,map)
title('Segment : 5')
%% Cyan portion
for i = 1:362
    for j = 1:434
        if file(i,j) >= 20 && file(i,j) <= 29
        mat(i,j) = 102;
        matrix(i,j) = mat(i,j);
        else
        mat(i,j) = 0;
        end
    end
end
map = jet;
subplot(2,3,6)
imshow(mat,map)
title('Segment : 6')
sgtitle(sprintf('Slice : %d',m))
% Labels:
Lab = imread(Labels_file(m));
for i = 1:362
    for j = 1:434
        if Lab(i,j) == 0
           Lab(i,j) = 1;
        end
    end
end
% Displaying the segmented image and the label
figure;
subplot(1,2,1)
imshow(matrix,map)
title('Segmented Image')
subplot(1,2,2)
imshow(Lab,map)
title('Labelled Image')
sgtitle(sprintf('Slice : %d',m))
% Checking dice score:
Label_1double = double(Lab);
matrix = double(matrix);
similarity = dice(matrix, Label_1double);
similarity2 = generalizedDice(matrix, Label_1double);

background(m) = similarity(1);
blue(m) = similarity(51);
cyan(m) = similarity(102);
yellow(m) = similarity(153);
red(m) = similarity(204);
maroon(m) = similarity(255);
gen(m) = similarity2;
end
back = mean(background);
blues = mean(blue);
cyans = mean(cyan);
yellows = mean(yellow);
reds = mean(red);
maroons = mean(maroon);
gens = mean(gen);
disp(["Similarity value of Air (Background):" back]);
disp(["Similarity value of Skin/Scalp (Blue):" blues]);
disp(["Similarity value of Skull (Cyan):" cyans]);
disp(["Similarity value of CSF (Yellow):" yellows]);
disp(["Similarity value of Gray Matter (Red):" reds]);
disp(["Similarity value of White Matter (Maroon):" maroons]);
disp(["The Generalized Dice Score is :" gens]);
```

2. Multi-Level Thresholding

```matlab
%% Multilevel thresholding:
background = zeros(1,10);
blue = zeros(1,10);
cyan = zeros(1,10);
yellow = zeros(1,10);
red = zeros(1,10);
maroon = zeros(1,10);
gen = zeros(1,10);
for m = 1:10
file = imread(T1_file(m));
X = grayslice(file,6);
X = uint8(X);
for i = 1:362
    for j = 1:434
        if X(i,j) == 1
            X(i,j) = 153;
        elseif X(i,j) == 3
            X(i,j) = 255;
        elseif X(i,j) == 2
            X(i,j) = 204;
        elseif X(i,j) == 4
            X(i,j) = 102;
        elseif X(i,j) == 5
            X(i,j) = 51;
        elseif X(i,j) == 0
            X(i,j) = 1;
        end
    end
end
Lab = imread(Labels_file(m));
% Changing background values from 0 to 1 as dice takes index values from 1
for i = 1:362
    for j = 1:434
        if Lab(i,j) == 0
           Lab(i,j) = 1;
        end
    end
end
X = double(X);
Label_1double = double(Lab);
similarity = dice(X, Label_1double);
similarity2 = generalizedDice(X, Label_1double);

figure;
subplot(1,2,1)
imshow(X, map)
axis off
title('Segmented Image')
subplot(1,2,2)
imshow(Lab,map)
axis off
title('Labelled Image')
sgtitle(sprintf('Slice : %d',m))
impixelinfo

background(m) = similarity(1);
blue(m) = similarity(51);
cyan(m) = similarity(102);
yellow(m) = similarity(153);
red(m) = similarity(204);
maroon(m) = similarity(255);
gen(m) = similarity2;
end
back = mean(background);
blues = mean(blue);
cyans = mean(cyan);
yellows = mean(yellow);
reds = mean(red);
maroons = mean(maroon);
gens = mean(gen);
disp(["Similarity value of Air (Background):" back]);
disp(["Similarity value of Skin/Scalp (Blue):" blues]);
disp(["Similarity value of Skull (Cyan):" cyans]);
disp(["Similarity value of CSF (Yellow):" yellows]);
disp(["Similarity value of Gray Matter (Red):" reds]);
disp(["Similarity value of White Matter (Maroon):" maroons]);
disp(["The Generalized Dice Score is :" gens]);
```

3. KMeans

```matlab
%% Implementing Kmeans:
background = zeros(1,10);
blue = zeros(1,10);
cyan = zeros(1,10);
yellow = zeros(1,10);
red = zeros(1,10);
maroon = zeros(1,10);
gen = zeros(1,10);

for m = 1:10
file = imread(T1_file(m));
[L,~] = imsegkmeans(file,6,NumAttempts=6);
% Correcting the labels to match the labels provided in order to analyse them.
if m == 1
    for i = 1:362
    for j = 1:434
        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;
        end
    end
    end
elseif m == 2
    for i = 1:362
    for j = 1:434
        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) = 1;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;
        end
    end
    end
elseif m == 3
    for i = 1:362
    for j = 1:434
        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;
        end
    end
    end
elseif m == 4
    for i = 1:362
    for j = 1:434
        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) = 204;elseif L(i,j) == 6;L(i,j) = 102;
        end
    end
    end
elseif m == 5
    for i = 1:362
    for j = 1:434
        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;
        end
    end
    end
elseif m == 6
    for i = 1:362
    for j = 1:434
        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) = 204;elseif L(i,j) == 6;L(i,j) = 102;
        end %cb mc br
    end
    end
elseif m == 7
    for i = 1:362
    for j = 1:434
        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;
        end
    end
```

```matlab
    end
elseif m == 8
    for i = 1:362
    for j = 1:434
        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) = 204;elseif L(i,j) == 5 ;L(i,j) = 1;elseif L(i,j) == 6;L(i,j) = 102;
        end
    end
    end
elseif m == 9
    for i = 1:362
    for j = 1:434
        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;
        end
    end
    end
elseif m == 10
    for i = 1:362
    for j = 1:434
        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) = 1;elseif L(i,j) == 4 ;L(i,j) = 255;elseif L(i,j) == 5 ;L(i,j) = 153;elseif L(i,j) == 6;L(i,j) = 102;
        end
    end
    end
end

Lab = imread(Labels_file(m));
% Changing Labels background values from 0 to 1 as dice takes index values from 1
for i = 1:362
    for j = 1:434
        if Lab(i,j) == 0
           Lab(i,j) = 1;
        end
    end
end
L = modefilt(L, [5,5]);
L = double(L);
%Converting to double in order to analyse
Label_1double = double(Lab);
similarity = dice(L, Label_1double);
similarity2 = generalizedDice(L, Label_1double);

figure;
map = jet;
subplot(1,2,1)
imshow(L, map)
impixelinfo
axis off
title("Segmented Image")

subplot(1,2,2)
imshow(Lab, map)
impixelinfo
title("Labelled Image")
sgtitle(sprintf('Slice : %d',m))

background(m) = similarity(1);
blue(m) = similarity(51);
cyan(m) = similarity(102);
gen(m) = similarity2;
end
back = mean(background);
blues = mean(blue);
cyans = mean(cyan);
yellows = mean(yellow);
reds = mean(red);
maroons = mean(maroon);
gens = mean(gen);
disp(["Similarity value of Air (Background):" back]);
disp(["Similarity value of Skin/Scalp (Blue):" blues]);
disp(["Similarity value of Skull (Cyan):" cyans]);
disp(["Similarity value of CSF (Yellow):" yellows]);
disp(["Similarity value of Gray Matter (Red):" reds]);
disp(["Similarity value of White Matter (Maroon):" maroons]);
disp(["The Generalized Dice Score is :" gens]);

```

4. Flood-fill Technique

```matlab
%% Implementing Flood-Fill Technique:
background = zeros(1,10);
blue = zeros(1,10);
cyan = zeros(1,10);
yellow = zeros(1,10);
red = zeros(1,10);
maroon = zeros(1,10);
gen = zeros(1,10);
for m = 1:10
img = imread(T1_file(m));
Mat = zeros(362,434);

J = grayconnected(img,34,34); % BACKGROUND
J = J*1;
J = uint8(J);
for i = 1:362
    for j = 1:434
        if J(i,j) == 1
            Mat(i,j) = J(i,j);
        end
    end
end
K = grayconnected(img,180,22); % BLUE
K = K*51;
K = uint8(K);
for i = 1:362
    for j = 1:434
        if K(i,j) == 51
            Mat(i,j) = K(i,j);
        end
    end
end
Q = grayconnected(img,96,76); % LIGHT BLUE
Q = Q*102;
Q = uint8(Q);
for i = 1:362
    for j = 1:434
        if Q(i,j) == 102
            Mat(i,j) = Q(i,j);
        end
    end
end
N = grayconnected(img,144,64); % RED
N = N*204;
N = uint8(N);
for i = 1:362
    for j = 1:434
        if N(i,j) == 204
            Mat(i,j) = N(i,j);
        end
    end
end
X = grayconnected(img,234,222); % MAROON
X = X*255;
X = uint8(X);
for i = 1:362
    for j = 1:434
        if X(i,j) == 255
            Mat(i,j) = X(i,j);
        end
    end
```

```
end

I = grayconnected(img,172,178); % YELLOW
I = I*153;
I = uint8(I);
for i = 1:362
    for j = 1:434
        if I(i,j) == 153
            Mat(i,j) = I(i,j);
        end
    end
end

% Changing background values from 0 to 1 as dice takes index values from 1
Lab = imread(Labels_file(m));
for i = 1:362
    for j = 1:434
        if Lab(i,j) == 0
           Lab(i,j) = J(i,j);
        end
    end
end

% Converting to double in order to analyse
Label_1double = double(Lab);
similarity = dice(Mat, Label_1double);
similarity2 = generalizedDice(Mat, Label_1double);

map = jet;
figure
subplot(1,2,1)
imshow(Mat, map);
impixelinfo;
title(sprintf('Segmented Image - Slice : %d',m))
%title(['The Generalized Dice Score is = ' num2str(similarity2)]);

subplot(1,2,2)
imshow(Lab, map)
impixelinfo;
title(sprintf('Labelled Image - Slice : %d',m))

background(m) = similarity(1);
blue(m) = similarity(51);
cyan(m) = similarity(102);
yellow(m) = similarity(153);
red(m) = similarity(204);
maroon(m) = similarity(255);
gen(m) = similarity2;
end
back = mean(background);
blues = mean(blue);
cyans = mean(cyan);
yellows = mean(yellow);
reds = mean(red);
maroons = mean(maroon);
gens = mean(gen);
disp(["Similarity value of Air (Background):" back]);
disp(["Similarity value of Skin/Scalp (Blue):" blues]);
disp(["Similarity value of Skull (Cyan):" cyans]);
disp(["Similarity value of CSF (Yellow):" yellows]);
disp(["Similarity value of Gray Matter (Red):" reds]);
```

5. Region Growing

```
background = zeros(1,10);
blue = zeros(1,10);
cyan = zeros(1,10);
yellow = zeros(1,10);
red = zeros(1,10);
maroon = zeros(1,10);
gen = zeros(1,10);
for m = 1:10
% First doing it only for the background
I = im2double(imread(T1_file(m)));
x=50; y=50;
J = regiongrowing(I,x,y,0.1);
J = J*1;
J = uint8(J);
figure;
subplot(2,3,1)
imshow(J);
impixelinfo;
colormap('jet')
title('Background Filter')

Mat = zeros(362,434);

% Changing background values from 0 to 1 as dice takes index values from 1
for i = 1:362
    for j = 1:434
        if J(i,j) == 1
            Mat(i,j) = J(i,j);
        end
    end
end

% Doing it for the next section: Blue
x=252; y=36;
%if m==1; val = 0.3; elseif m ==2; val = 0.3; end
D = regiongrowing(I,x,y,0.2);
D = D*51;
D = uint8(D);
%K = imadjust(I,[0 1]);
subplot(2,3,2)
imshow(D);
colormap('jet')
title('Blue Filter')

for i = 1:362
    for j = 1:434
        if D(i,j) == 51
            Mat(i,j) = D(i,j);
        end
    end
end

% Doing it for the next section: Light Blue
x=74; y=94;
X = regiongrowing(I,x,y,0.09);
X = X*102;
X = uint8(X);
%K = imadjust(I,[0 1]);
subplot(2,3,3)
imshow(X);
colormap('jet')
```

```
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 = N*204;
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')
```

```
sgtitle(sprintf('Slice : %d',m))

for i = 1:362
    for j = 1:434
        if M(i,j) == 153
            Mat(i,j) = M(i,j);
        end
    end
end

%% Computing Generalized Dice Score:
% Convering to double in order to analyse
Lab = imread(Labels_file(m));
for i = 1:362
    for j = 1:434
        if Lab(i,j) == 0
           Lab(i,j) = J(i,j);
        end
    end
end
Label_1double = double(Lab);
similarity = dice(Mat, Label_1double);
similarity2 = generalizedDice(Mat, Label_1double);

map = jet;
figure
subplot(1,2,1)
imshow(Mat, map);
impixelinfo;
title('Segmented Image')
%title(['The Generalized Dice Score is = ' num2str(similarity)])

% Looking at the Labels
subplot(1,2,2)
imshow(Lab, map);
impixelinfo;
title('Original Labels')
sgtitle(sprintf('Slice : %d',m))

background(m) = similarity(1);
blue(m) = similarity(51);
cyan(m) = similarity(102);
yellow(m) = similarity(153);
red(m) = similarity(204);
maroon(m) = similarity(255);
gen(m) = similarity2;
end
back = mean(background);
blues = mean(blue);
cyans = mean(cyan);
yellows = mean(yellow);
reds = mean(red);
maroons = mean(maroon);
gens = mean(gen);
disp(["Similarity value of Air (Background):" back]);
disp(["Similarity value of Skin/Scalp (Blue):" blues]);
disp(["Similarity value of Skull (Cyan):" cyans]);
disp(["Similarity value of CSF (Yellow):" yellows]);
disp(["Similarity value of Gray Matter (Red):" reds]);
disp(["Similarity value of White Matter (Maroon):" maroons]);
disp(["The Generalized Dice Score is :" gens]);
```

```
function J=regiongrowing(I,x,y,reg_maxdist)
if(exist('reg_maxdist','var')==0), reg_maxdist=0.2; end
if(exist('y','var')==0), figure, imshow(I,[]); [y,x]=getpts; y=round(y(1)); x=round(x(1)); end

J = zeros(size(I));
Isizes = size(I);

reg_mean = I(x,y); % The mean of the segmented region
reg_size = 1;

neg_free = 10000; neg_pos=0;
neg_list = zeros(neg_free,3);

pixdist=0;
neigb=[-1 0; 1 0; 0 -1;0 1];

while(pixdist<reg_maxdist&&reg_size<numel(I))

    for j=1:4
        xn = x +neigb(j,1); yn = y +neigb(j,2);
        ins=(xn>=1)&&(yn>=1)&&(xn<=Isizes(1))&&(yn<=Isizes(2));
        if(ins&&(J(xn,yn)==0))
                neg_pos = neg_pos+1;
                neg_list(neg_pos,:) = [xn yn I(xn,yn)]; J(xn,yn)=1;
        end
    end
    if(neg_pos+10>neg_free), neg_free=neg_free+10000; neg_list((neg_pos+1):n
    dist = abs(neg_list(1:neg_pos,3)-reg_mean);
    [pixdist, index] = min(dist);
    J(x,y)=2; reg_size=reg_size+1;
    reg_mean= (reg_mean*reg_size + neg_list(index,3))/(reg_size+1);
    x = neg_list(index,1); y = neg_list(index,2);
    neg_list(index,:)=neg_list(neg_pos,:); neg_pos=neg_pos-1;
end
J=J>1;
```

6. Combined 2D

```
%% COMBINED 2D MODEL:
background = zeros(1,10);
blue = zeros(1,10);
cyan = zeros(1,10);
yellow = zeros(1,10);
red = zeros(1,10);
maroon = zeros(1,10);
gen = zeros(1,10);
mat2 = zeros(362,434);
mat4 = zeros(362,434);
for m = 1:10
%img = imread(T1_file(m));
file = imread(T1_file(m));
filedoub = im2double(file);
file2 = medfilt2(file,[9 9]);
file2doub = im2double(file2);
Mat = zeros(362,434);

%% BACKGROUND: AIR
x=50; y=50;
J = regiongrowing(filedoub,x,y,0.1);
J = J*1;
J = uint8(J);
Mat = zeros(362,434);

% Changing background values from 0 to 1 as dice takes index values from 1
for i = 1:362
    for j = 1:434
        if J(i,j) == 1
            Mat(i,j) = J(i,j);
        end
    end
end

%% CYAN
x=74; y=94;
X = regiongrowing(filedoub,x,y,0.09);
X = X*102;
X = uint8(X);
for i = 1:362
    for j = 1:434
        if X(i,j) == 102
            Mat(i,j) = X(i,j);
        end
    end
end

%% RED
for i = 1:362
    for j = 1:434
        if file(i,j) >= 115 && file(i,j) <= 120
        mat4(i,j) = 204;
        else
        mat4(i,j) = 0;
        end
    end
end
% Doing it for the next section: Red
x=194; y=201;
if m == 10; val = 0.2; else val = 0.09; end
N = N*204;
N = uint8(N);
for i = 1:362
    for j = 1:434
        if N(i,j) == 204
            mat4(i,j) = N(i,j);
            Mat(i,j) = mat4(i,j);
        end
    end
end

%% MAROON
[L,~] = imsegkmeans(file,6,NumAttempts=6);
% Correcting the labels to match the labels provided in order to analyse them
if m == 1
    for i = 1:362
    for j = 1:434
        if L(i,j) == 1 ; L(i,j) = 255; Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 2
    for i = 1:362
    for j = 1:434
        if L(i,j) == 5; L(i,j) = 255;Mat(i,j) =  L(i,j);
disp(["The Generalized Dice Score is :" num2str(similarity2)]);
    end
    end
elseif m == 3
    for i = 1:362
    for j = 1:434
        if L(i,j) == 5; L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 4
    for i = 1:362
    for j = 1:434
        if L(i,j) == 2;L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 5
    for i = 1:362
    for j = 1:434
        if L(i,j) == 4 ;L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 6
    for i = 1:362
    for j = 1:434
        if L(i,j) == 3;L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 7
    for i = 1:362
    for j = 1:434
        if L(i,j) == 3;L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 8
    for i = 1:362
    for j = 1:434
        if L(i,j) == 2; L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 9
    for i = 1:362
    for j = 1:434
        if L(i,j) == 6;L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
elseif m == 10
    for i = 1:362
    for j = 1:434
        if L(i,j) == 4 ;L(i,j) = 255;Mat(i,j) =  L(i,j);
        end
    end
    end
end

%% YELLOW
X = grayslice(file,6);
X = uint8(X);
for i = 1:362
    for j = 1:434
        if X(i,j) == 1
            X(i,j) = 153;
            Mat(i,j) = X(i,j);
        end
    end
end

%% BLUE
for i = 1:362
    for j = 1:434
        if file2(i,j) >= 170 && file2(i,j) <= 255
        mat2(i,j) = 51;
        else
        mat2(i,j) = 0;
        end
    end
end
x=73; y=68;
D = regiongrowing(file2doub,x,y,0.18);
D = D*51;
D = uint8(D);
for i = 1:362
    for j = 1:434
        if D(i,j) == 51
            mat2(i,j) = D(i,j);
            Mat(i,j) = mat2(i,j);
        end
    end
end

Mat = modefilt(Mat,[5 5]);

% Changing background values from 0 to 1 as dice takes index values from 1
Lab = imread(Labels_file(m));
for i = 1:362
    for j = 1:434
        if Lab(i,j) == 0
            Lab(i,j) = J(i,j);
        end
    end
end

% Converting to double in order to analyse
Label_1double = double(Lab);
similarity = dice(Mat, Label_1double);
similarity2 = generalizedDice(Mat, Label_1double);

map = jet;
figure
subplot(1,2,1)
imshow(Mat, map);
impixelinfo;
title(sprintf('Segmented Image - Slice : %d',m))
%title(['The Generalized Dice Score is = ' num2str(similarity2)]);

subplot(1,2,2)
imshow(Lab, map)
impixelinfo;
title(sprintf('Labelled Image - Slice : %d',m))

background(m) = similarity(1);
blue(m) = similarity(51);
cyan(m) = similarity(102);
yellow(m) = similarity(153);
red(m) = similarity(204);
maroon(m) = similarity(255);
gen(m) = similarity2;
end
back = mean(background);
blues = mean(blue);
cyans = mean(cyan);
yellows = mean(yellow);
reds = mean(red);
maroons = mean(maroon);
gens = mean(gen);
disp(["Similarity value of Air (Background):" back]);
disp(["Similarity value of Skin/Scalp (Blue):" blues]);
disp(["Similarity value of Skull (Cyan):" cyans]);
disp(["Similarity value of CSF (Yellow):" yellows]);
disp(["Similarity value of Gray Matter (Red):" reds]);
disp(["Similarity value of White Matter (Maroon):" maroons]);
disp(["The Generalized Dice Score is :" gens]);
```

7. KMeans 3D

```
volshow(T1)
% Segmenting the volume into 6 clusters
L = imsegkmeans3(T1,6, MaxIterations = 100);
for i = 1:362
    for j = 1:434
        for m = 1:10
        if L(i,j,m) == 2
            L(i,j,m) = 255;
        elseif L(i,j,m) == 3
            L(i,j,m) = 1;
        elseif L(i,j,m) == 1
            L(i,j,m) = 153;
        elseif L(i,j,m) == 4
            L(i,j,m) = 204;
        elseif L(i,j,m) == 5
            L(i,j,m) = 51;
        elseif L(i,j,m) == 6
            L(i,j,m) = 102;
        end
        end
    end
end
figure
L = modefilt(L, [5,5,5]);
volshow(L);
% Changing the label value to
labels = uint8(255*mat2gray(label(:,:,:)));
for i = 1:362
    for j = 1:434
        for m = 1:10
        if labels(i,j,m) == 0
           labels(i,j,m) = 1;
        end
        end
    end
end
for m = 1:10
figure
subplot(1,2,1)
map = jet;
imshow(L(:,:,m), map)
title('Segmented Image')
subplot(1,2,2)
map = jet;
imshow(labels(:,:,m), map)
title('Labelled Image')
sgtitle(sprintf('Slice : %d',m))
end

% Calculating the overall dice value
Ldouble = double(L);
Label_1double = double(labels);
similarity = dice(Ldouble, Label_1double);
similarity2 = generalizedDice(Ldouble, Label_1double);

disp(["Similarity value of Air (Background):" similarity(1)]);
disp(["Similarity value of Skin/Scalp (Blue):" similarity(51)]);
disp(["Similarity value of Skull (Cyan):" similarity(102)]);
disp(["Similarity value of CSF (Yellow):" similarity(153)]);
disp(["Similarity value of Gray Matter (Red):" similarity(204)]);
disp(["Similarity value of White Matter (Maroon):" similarity(255)]);
```

8. Thresholding3D

```
%% Thresholding3D
figure
volshow(T1)
conv = uint8(255*mat2gray(T1));
file = medfilt3(file, [5,5,5] );
file2 = medfilt3(file, [1,1,1]);
mat1 = zeros(362,434,10);mat2 = zeros(362,434,10);mat3 = zeros(362,434,10);
mat4 = zeros(362,434,10);mat5 = zeros(362,434,10);mat6 = zeros(362,434,10);
matrix = zeros(362,434,10);
figure
imhist(conv)

%% Background
for i = 1:362
    for j = 1:434
        for m = 1:10
        if file(i,j,m) >= 0 && file(i,j,m) <= 15
        mat1(i,j,m) = 1;
        matrix(i,j,m) = mat1(i,j,m);
        else
        mat1(i,j,m) = 0;
        end
        end
    end
end
%% Yellow portion
for i = 1:362
    for j = 1:434
        for m = 1:10
        if conv(i,j,m) >= 30 && conv(i,j,m) <= 55
        mat3(i,j,m) = 153;
        matrix(i,j,m) = mat3(i,j,m);
        else
        mat3(i,j,m) = 0;
        end
        end
    end
end
%% Red portion
for i = 1:362
    for j = 1:434
        for m = 1:10
        if conv(i,j,m) >= 80 && conv(i,j,m) <= 130
        mat4(i,j,m) = 204;
        matrix(i,j,m) = mat4(i,j,m);
        else
        mat4(i,j,m) = 0;
        end
        end
    end
end
%% Maroon portion
for i = 1:362
    for j = 1:434
        for m = 1:10
        if conv(i,j,m) >= 130 && conv(i,j,m) <= 170
        mat5(i,j,m) = 255;
        matrix(i,j,m) = mat5(i,j,m);
        else
        mat5(i,j,m) = 0;
        end
        end
    end
end
%% Cyan portion
for i = 1:362
    for j = 1:434
        for m = 1:10
        if file2(i,j,m) >= 20 && file2(i,j,m) <= 29
        mat6(i,j,m) = 102;
        matrix(i,j,m) = mat6(i,j,m);
        else
        mat6(i,j,m) = 0;
        end
        end
    end
end
%% Blue
for i = 1:362
    for j = 1:434
        for m = 1:10
        if conv(i,j,m) >= 170 && conv(i,j,m) <= 255
        mat2(i,j,m) = 51;
        matrix(i,j,m) = mat2(i,j,m);
        else
        mat2(i,j,m) = 0;
        end
        end
    end
end
matrix = modefilt(matrix, [5,5,5]);
volshow(matrix)


% Calculating the overall dice value
Ldouble = double(matrix);
Label_1double = double(labels);
similarity = dice(Ldouble, Label_1double);
similarity2 = generalizedDice(Ldouble, Label_1double);

disp(["Similarity value of Air (Background):" similarity(1)]);
disp(["Similarity value of Skin/Scalp (Blue):" similarity(51)]);
disp(["Similarity value of Skull (Cyan):" similarity(102)]);
disp(["Similarity value of CSF (Yellow):" similarity(153)]);
disp(["Similarity value of Gray Matter (Red):" similarity(204)]);
disp(["Similarity value of White Matter (Maroon):" similarity(255)]);
disp(["The Generalized Dice Score is :" num2str(similarity2)]);
```

9. Combined 3D

```
figure
volshow(T1)
conv = uint8(255*mat2gray(T1));
file = medfilt3(conv, [5,5,5]);
mat1 = zeros(362,434,10);mat2 = zeros(362,434,10);mat3 = zeros(362,434,10);
mat4 = zeros(362,434,10);mat5 = zeros(362,434,10);mat6 = zeros(362,434,10);
matrix = zeros(362,434,10);
figure
imhist(conv)

%% Background
for i = 1:362
    for j = 1:434
        for m = 1:10
        if file(i,j,m) >= 0 && file(i,j,m) <= 15
        mat1(i,j,m) = 1;
        matrix(i,j,m) = mat1(i,j,m);
        else
        mat1(i,j,m) = 0;
        end
        end
    end
end
%% Blue
L = imsegkmeans3(T1,6, MaxIterations = 100);
for i = 1:362
    for j = 1:434
        for m = 1:10
        if L(i,j,m) == 5
            L(i,j,m) = 51;
            matrix(i,j,m) = L(i,j,m);
        end
        end
    end
end

%% Cyan portion
for i = 1:362
    for j = 1:434
        for m = 1:10
        if file(i,j,m) >= 20 && file(i,j,m) <= 30
        mat6(i,j,m) = 102;
        matrix(i,j,m) = mat6(i,j,m);
        else
        mat6(i,j,m) = 0;
        end
        end
    end
end

%% Yellow
L = imsegkmeans3(T1,6, MaxIterations = 100);
for i = 1:362
    for j = 1:434
        for m = 1:10
        if  L(i,j,m) == 1
            L(i,j,m) = 153;
            matrix(i,j,m) = L(i,j,m);
        end
        end
    end
end

%% Red
L = imsegkmeans3(T1,6, MaxIterations = 100);
for i = 1:362
    for j = 1:434
        for m = 1:10
        if L(i,j,m) == 4
           L(i,j,m) = 204;
           matrix(i,j,m) = L(i,j,m);
        end
        end
    end
end

%% Maroon
L = imsegkmeans3(T1,6, MaxIterations = 100);
for i = 1:362
    for j = 1:434
        for m = 1:10
        if L(i,j,m) == 2
            L(i,j,m) = 255;
            matrix(i,j,m) = L(i,j,m);

        end
        end
    end
end
matrix = modefilt(matrix, [5,5,5]);
volshow(matrix)


% Calculating the overall dice value
Ldouble = double(matrix);
Label_1double = double(labels);
similarity = dice(Ldouble, Label_1double);
similarity2 = generalizedDice(Ldouble, Label_1double);

disp(["Similarity value of Air (Background):" similarity(1)]);
disp(["Similarity value of Skin/Scalp (Blue):" similarity(51)]);
disp(["Similarity value of Skull (Cyan):" similarity(102)]);
disp(["Similarity value of CSF (Yellow):" similarity(153)]);
disp(["Similarity value of Gray Matter (Red):" similarity(204)]);
disp(["Similarity value of White Matter (Maroon):" similarity(255)]);
disp(["The Generalized Dice Score is :" num2str(similarity2)]);
```