

BML 735
BIOMEDICAL SIGNAL & IMAGE PROCESSING

Automatic segmentation of tissues

PROJECT REPORT



**INDIAN INSTITUTE OF TECHNOLOGY,
DELHI**

Instructor:
Prof. Anup Singh

Submitted By -
Ajinkya Wasnik
2018EET2569

Automatic Segmentation of tumor tissues in Brain.

Problem Statement:

Develop implement an algorithm for Segmentation of tumor tissues (contrast enhancing and non-contrast enhancing components) based upon thresholding, morphological operations, k-mean clustering. For contrast enhancing component use 'Brats18_2013_2_1_t1ce.nii' images. For non-contrast enhancing you can use FLAIR or T2- W images.

- Compute **accuracy** of your algorithm using **Dice coefficient** w.r.t. gold standard mask 'Brats18_2013_2_1_seg.nii'. It should be sufficiently high (>75%).
 - **Compare** this algorithm results with **region growing** based algorithm
 - Perform following analysis on segmented tumor tissue components:
volume, mean intensity, SD, histogram.
 - You have to **test** your segmentation algorithm on **three different patient data** (BRATS data).
-

1.Introduction:

Each patient data contains following images (each having multiple images of entire tumor):

1. Brats18_2013_2_1_flair.nii or FLAIR weighted images
2. Brats18_2013_2_1_seg.nii or Mask
3. Brats18_2013_2_1_t1.nii or T1-W images
4. Brats18_2013_2_1_t1ce.nii or T1 CE images
5. Brats18_2013_2_1_t2.nii or T2 Weighted images

I have considered the '*flair.nii*' and '*t1ce.nii*' for this purpose of segmenting tumorous and non-tumorous tissues.

****Note: All screenshots attached are of patient 7 unless stated otherwise**

2. Objective:

The objective of this project is to investigate the use of thresholding, morphological operations, clustering operations for distinguishing different tumorous brain tissues and segment them, while calculating the dice coefficient achieved in the process of segmentation and also to find volume, mean intensity, standard deviation and histogram of different tumorous tissues.

3. Performance Metrics:

Dice coefficient, mean, standard deviation on the masks obtained for various tumorous tissue of the classifiers are tabulated.

4. Exploratory Data Analysis:

Preprocessing: The data was provided as separate zip files which had those patient's folders wherein the five types of images named earlier are present. We unzip them all normally or in Matlab using command `unzip ()` so that *.nii* images could be read.

5. Methods:

It's well known to us that there are various types of segmentation methods:

- i) Thresholding method
- ii) Region Based
- iii) Clustering Based
- iv) Partial differential equation based
- v) Watershed based techniques
- vi) Artificial neural network based techniques, etc.

I've tried the first three methods for this project and the methodology and steps are explained in depth in subsequent points.

6.Step by Step going through the Algorithm developed for segmenting tumor:

Step1:

Taking the path for the patients *t1ce* , *flair*, and its respective mask named *seg.nii*.

Matlab code snippet:

```
%directories selection for patient's t1ce, seg.nii, flair

dirt1ce = '..\Data_BraTS2018\Brats18_2013_7_1\Brats18_2013_7_1_t1ce.nii';
dirgtmask='..\Data_BraTS2018\Brats18_2013_7_1\Brats18_2013_7_1_seg.nii';
dirfl='..\Data_BraTS2018\Brats18_2013_7_1\Brats18_2013_7_1_flair.nii';
```

Here as file name suggests I took patient 7 under consideration.

Step 2: (optional)

As my version of Matlab is 15a so it doesn't supports niftiread() function which came from 17b and subsequent versions. So for doing same thing in 15a , I had to get some .m files which do the same task for me which I've added in the archive submitted.

Matlab code snippet:

```
%directories dirt1ce for t1ce img
%t1c is t1 contrast enhanced image
t1c_nii = load_nii(dirt1ce);
t1c = t1c_nii.img;
```

OR

```
%If we have 17b or newer version
T1c=niftiread(dirt1ce);
*Similarly we load .nii images for its mask and flair image
```

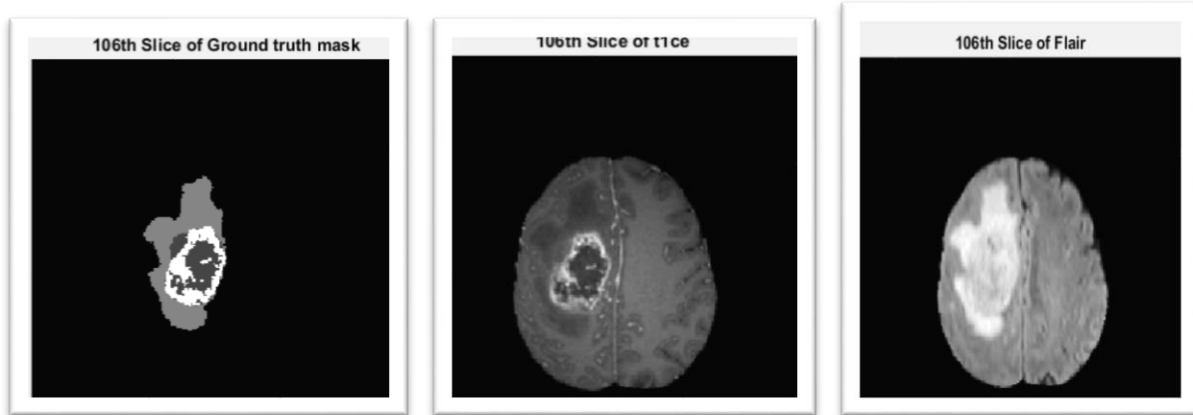
*Showing some slice of the image loaded of all t1ce, flair, mask

Matlab code snippet:

```
% showing 106th slice of each image loaded
figure, imagesc(t1c(:,:,106)),colormap(gray),axis image,axis off,title('106th Slice of t1ce');

figure, imagesc(fl(:,:,106)),colormap(gray),axis image,axis off,title('106th Slice of Flair');

figure, imagesc(gtmask(:,:,106)),colormap(gray),axis image,axis off,title('106th Slice of Ground truth mask');
```



Step 3: Algorithm Begins: Now we start by taking one slice of each image and apply following steps sequentially for achieving main purpose.

a) Apply operations only if non-zero no. of pixels in that particular ground truth is greater than 1000. (**Thresholding** in a sense) also, checking if max value of ground truth is less than 4.

Matlab code snippet:

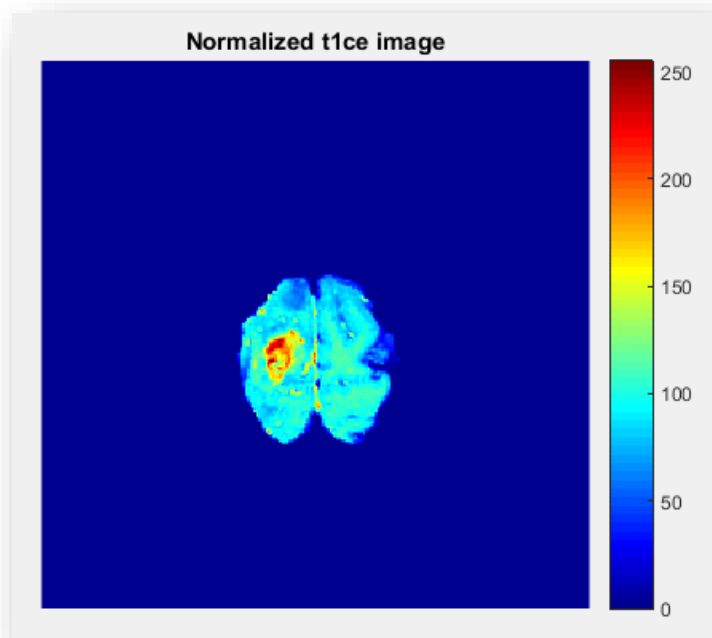
```
% Applying thresholding of non-zero no. of pixels in ground truth and also
associated max. value of ground truth
for i = 1:155                                     %for all slices check
    T1 = t1c(:,:,i);
    F1 = f1(:,:,i);
    gImg =gtmask(:,:,i);

    if max(gImg)<4 | nnz(gImg(:))<1000
        continue;
    end
```

b) Normalizing

Matlab code snippet:

```
T1N = uint8((double(T1-min(T1(:)))/double(max(T1(:))-min(T1(:))))*255);
```



c) Also creating different ground truth mask for different tumor tissues.

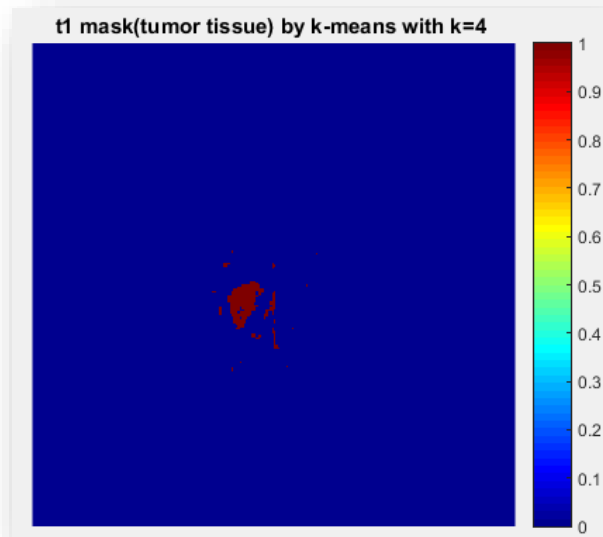
Matlab code snippet:

```
gImg1 (gImg (:)==1 | gImg (:)==2)=0;
gImgf (gImg (:)==1 | gImg (:)==4)=0;
gImge (gImg (:)==2 | gImg (:)==4)=0;
```

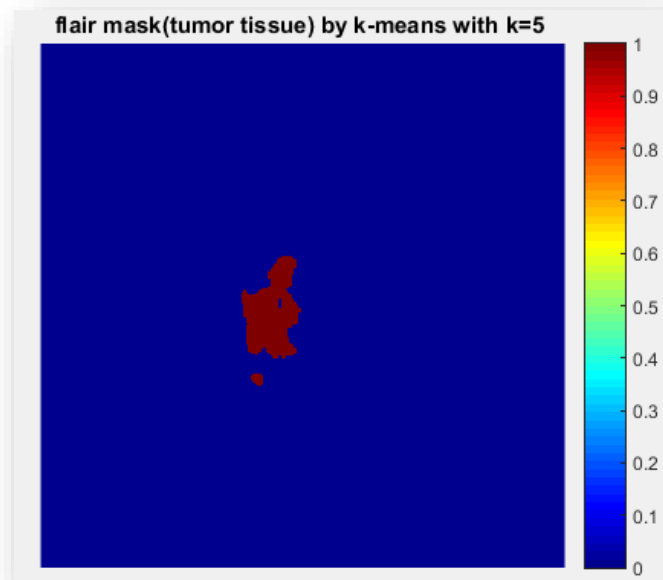
d) Applying K-means so that we get segmentation

Matlab code snippet:

```
k=4; k1=5;
[mut,maskt1]=kmeans1 (T1N,k);
```



```
[muf,maskf1]=kmeans1(F1N,k1);
```



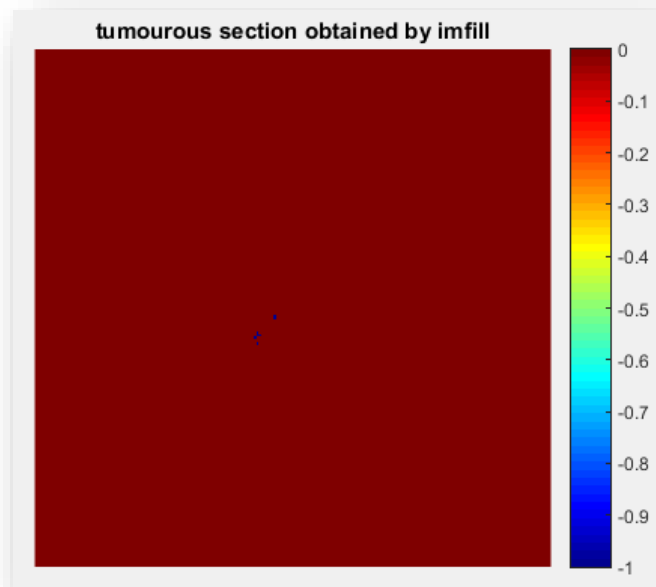
```
%k1 for getting different segmentation in F1N image's tumour tissues
```

```
Img_maskt1(maskt1(:)==4)=1;
```

```
%getting a third portion also by filling with region 4 and then subtracting  
it from original
```

```
Img_mask0=imfill(Img_maskt1,'holes');
```

```
Img_mask0=Img_maskt1-Img_mask0;
```

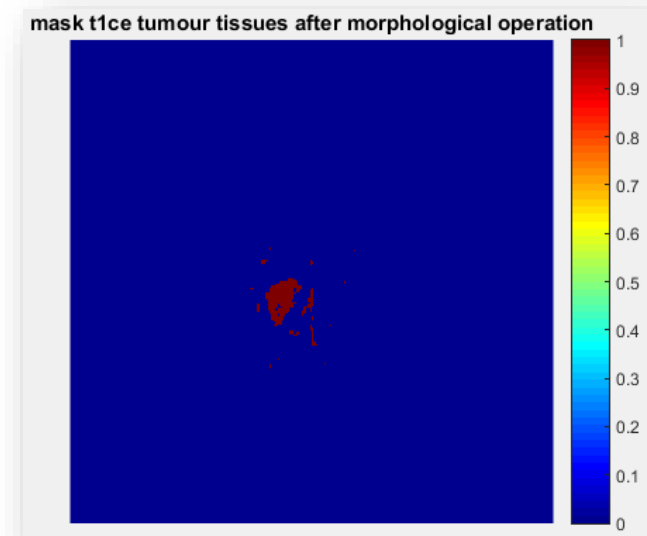


e) Morphological operation

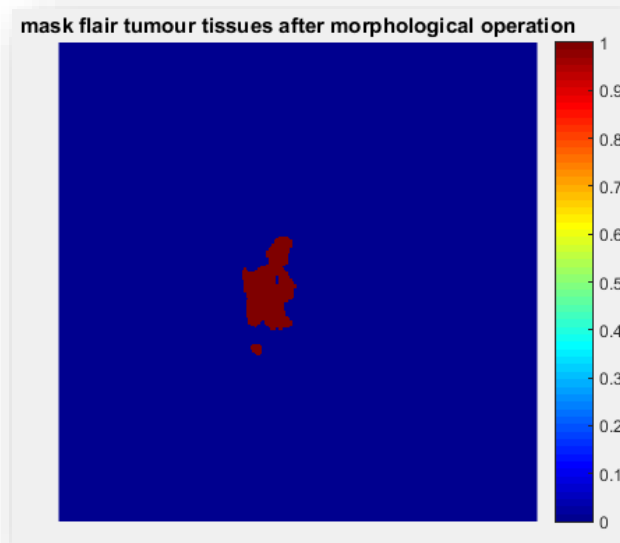
The resulting masks we get from step 3d) we apply opening operating to improve size of tumor mask.

Matlab code snippet:

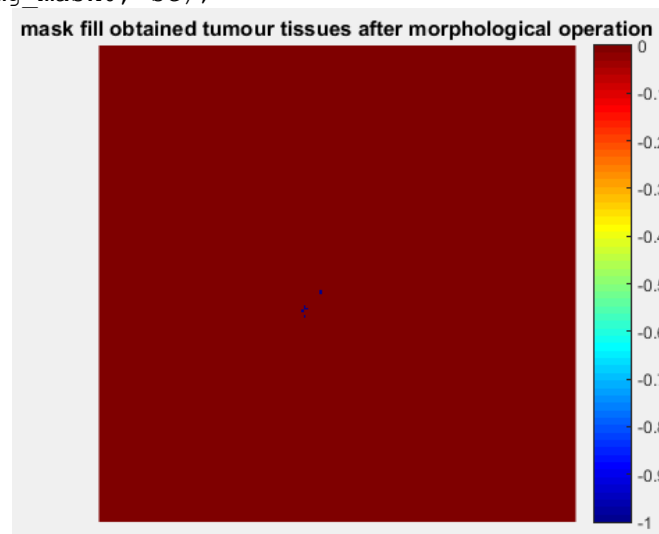
```
se = strel('square', 1);
maskt1 = imopen(Img_maskt1, se);
```



```
maskf1 = imopen(Img_maskf1, se);
```

```
mask1 = imopen(Img_mask0, se);
```

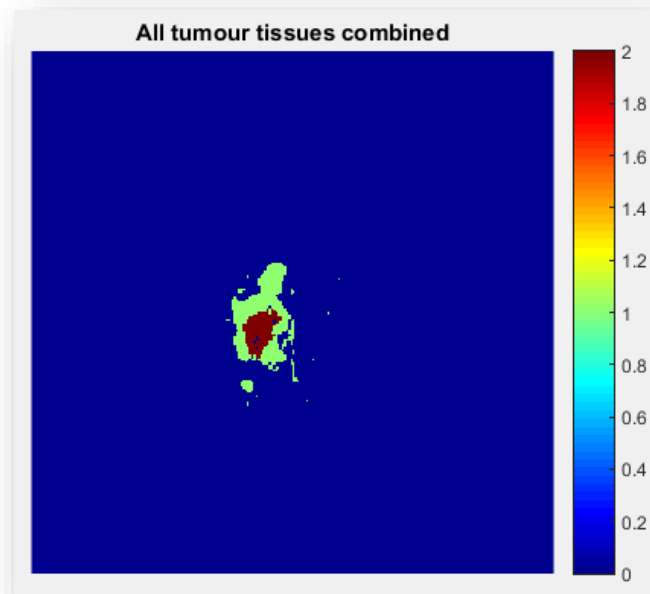


f) Calculation of Dice's Coefficient

We accumulate the no. of non-zeros of ANDing between ground truth mask and mask of tumors
And also no. of non-zeros of ground truth and mask of tumors in the same for $i=1:155$ loop
Mask of tumor here is the accumulation of all tumor types in the .nii file which we got by adding them eventually.

```
%% dice coefficeint computation
```

```
mask=maskel+maskfl+maskt1;
```



```
dand=dand+nnz(gImg & mask);  
dplus=dplus+nnz(gImg) + nnz(mask);
```

g) Placing the values eventually for calculating Dice's coefficient.

```
dice = 2*dand/dplus;
```

```
>> part1  
dice coefficient : 0.856654
```

dice	0.8567
------	--------

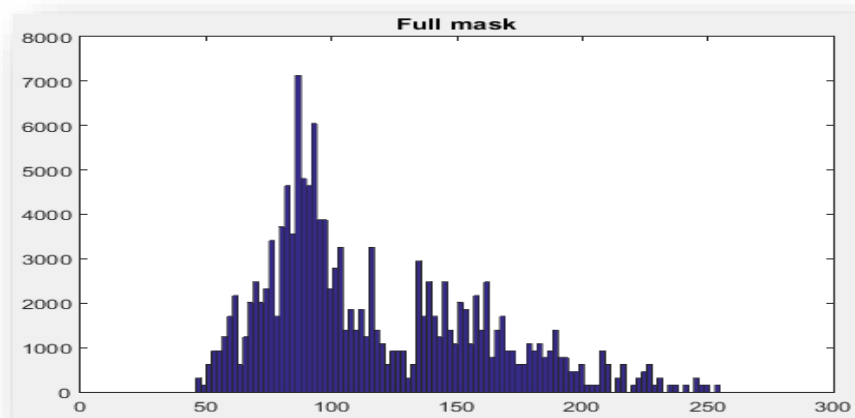
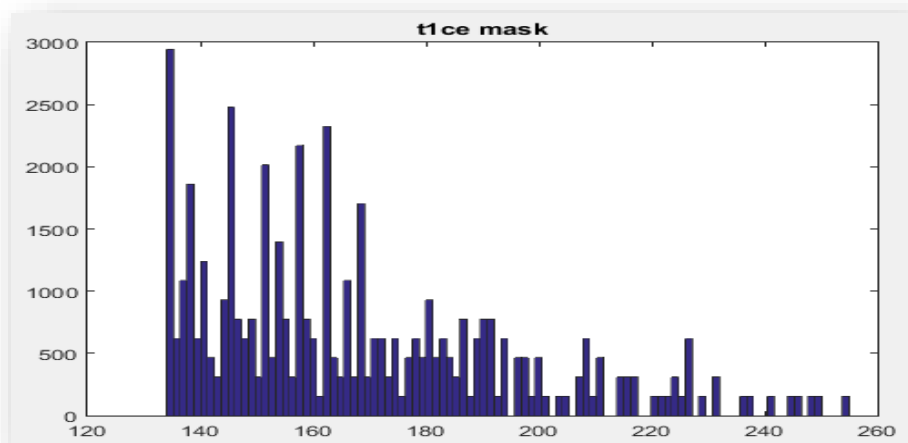
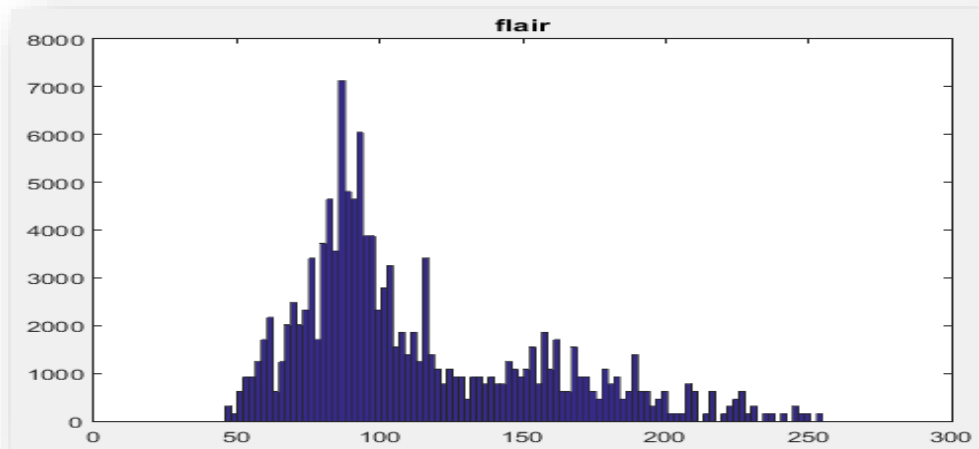
Step 4: Creating a segmented image corresponding to each mask of different tumor cells

Matlab code snippet:

```
segImg=zeros(240,240,155);  
for i=1:155  
    mask1=mask;  
    mask1(mask(:)>0)=1;mask1(mask(:)==0)=0;  
    segImg(:,:,i)=T1N.*uint8(mask);  
end
```

Step5: Showing **histogram** for those tumour segments

```
figure,hist(segImg(segImg>0),100),title('Full mask');
```



Step6: Calculating Mean intensities

```
meanIntensity=mean(nonzeros(segImg(:)));
```

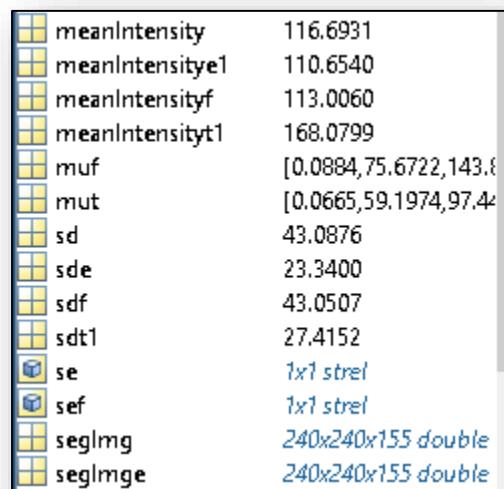
Step7: Calculating Standard Deviation

```
sd=std(segImg(segImg(:)>0));
```

Step8: Calculating Volume

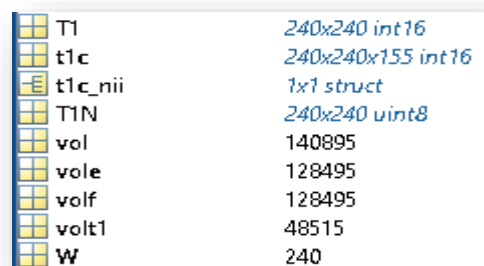
```
vol=nnz(segImg(:));
```

****Note: We have to do such calculations for all tumorous tissue segments.(done in code)(values below correspond to that code)**



meanIntensity	116.6931
meanIntensitye1	110.6540
meanIntensityf	113.0060
meanIntensityt1	168.0799
muf	[0.0884,75.6722,143.6]
mut	[0.0665,59.1974,97.4]
sd	43.0876
sde	23.3400
sdf	43.0507
sdt1	27.4152
se	1x1 struct
sef	1x1 struct
segImg	240x240x155 double
segImge	240x240x155 double

fig: Mean Intensity and Standard deviation values for different tumour tissues.



T1	240x240 int16
t1c	240x240x155 int16
t1c_nii	1x1 struct
T1N	240x240 uint8
vol	140895
vole	128495
volf	128495
volt1	48515
W	240

fig: Volume values for different tumour tissues.

7. Region Growing Image Segmentation.

Algorithmic steps for region growing

Step 1: let $C_1, C_2 \dots C_n$ be the n cluster which is no of clustered seed points $P_1, P_2 \dots P_n$ be the position of initial seed points.

Step 2: The difference of pixel value of the initial seed point P_i and its neighboring point is classified in $C_i, i=1, 2 \dots n$

Step 3: Re-compute the boundary of C_i and set those boundary points as new seed points $P_i(s)$. In addition, compute the respective mean pixel value of C_i .

Step 4: Step 2 and Step 3 is repeated until all the pixels in the image have be allocated to a suitable cluster.

****NOTE:** We wish to apply region growing on 3D data so I have used matlab code[1] .

Various hyper-parameters are required to be tuned to use that region growing.

cIM: 2D/3D grayscale matrix {current image}
initPos: Coordinates for initial seed position {ginput position}
thresVal: Absolute threshold level to be included {5% of max-min}
maxDist: Maximum distance to the initial position in [px] {Inf}
tfMean: Updates the initial value to the region mean (slow) {false}
tfFillHoles: Fills enclosed holes in the binary mask {true}
tfSimplify: Reduces the number of vertices {true, if dpsimplify exists}

I have taken value of threshold 6% of max-min value.

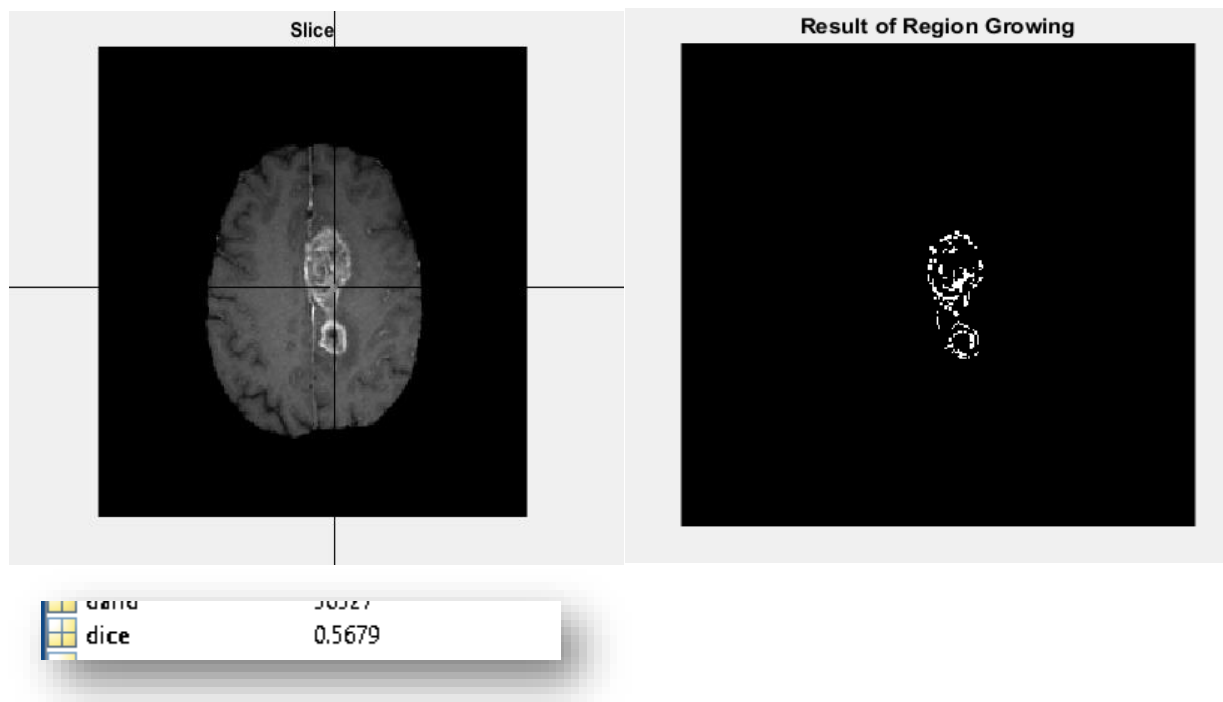
All other parameters suiting to context.

```
%considering d_mask has address of directory where the image to be segmented  
is present  
U_nii = load_nii(d_mask);  
U = U_nii.img;  
U = imrotate(U, 90);  
%for taking seed point  
[y, x] = ginput();
```

```

%maually also we can assign seed.
    x1 = 109
    y1 = 92;
x1=round(x);
y1=round(y);
thres = 0.06*(max(max(V(:,:,106)))- min(min(V(:,:,106))));
[polygon, mask] = rg(V,[x1,y1,106],thres, 130, false, true, false);
% index
figure,imagesc(mask(:,:,106)),colormap(gray),axis image,axis
off,title('Result of Region Growing');
dice=getDiceCoeff(U(:,:,106),mask(:,:,106)); % calculates dice coefficient

```



Dice coeff : 0.5679 (Values checked on patient2)

Comparison:

- i) Algorithm which I worked out works well and provides comparatively better dice coefficient value .
- ii) Region growing output varies a lot according to theselectedseed point which mightbe a curse a lot of times.
- iii) Computationally slow when the run time of both are compared i.e., Algorithm coded here by me and the region growing algorithm.
- iv)SD also varies in both algorithms.

8. Results

i) I took a tumor mask for T1ce and another one mask for flair and also a third tumor mask from the filling of 4th index section and subtracting it(Explained in previous sections with code)

ii) Final Results are combination of all the masks.

iii) Subject-wise the data varies in different metrics as shown in below table

So. Average value of Dice coefficient achieved considered 3 patients
= **0.834033 (around 84% accuracy achieved)**

Tissues	Mean Intensity	SD	Volume	Dice Coeff.
Subject 2	107.8348	27.5	141670	0.825182
Subject 7	116.69	43.0876	140895	0.856654
Subject 17	93.55	21.67	214985	0.820263

Table1: Table showing subject-wise statistics

Metrics	Tumor1	Tumor2	Tumor4	Overall
Volume	2889	121365	21080	141670
Mean	71.67	98.30054	110.2	107.8
s.d.	29.99	12.14	25.26	27.58

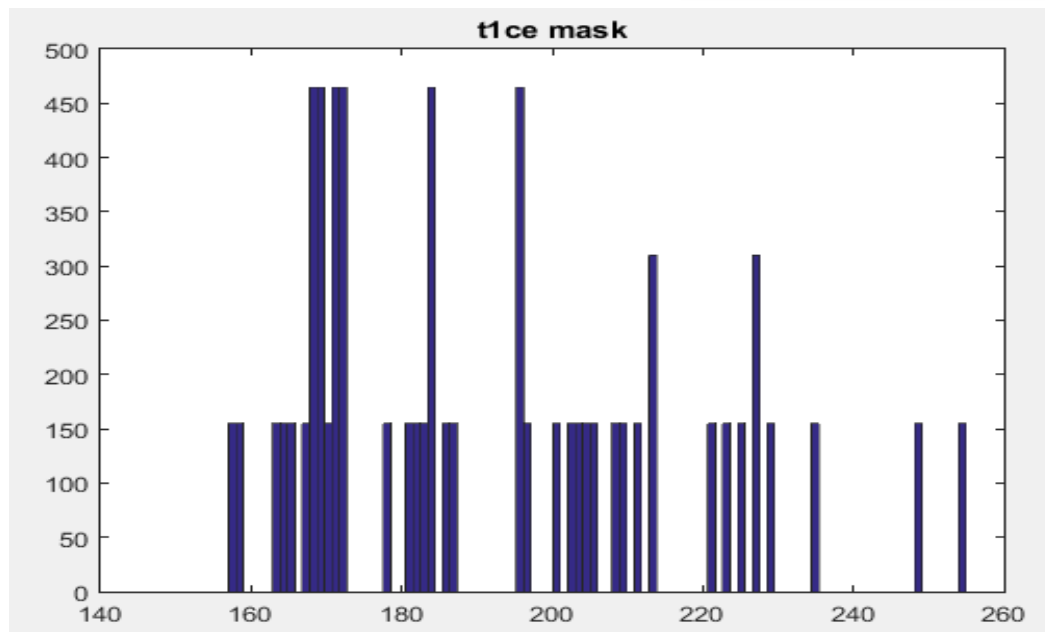
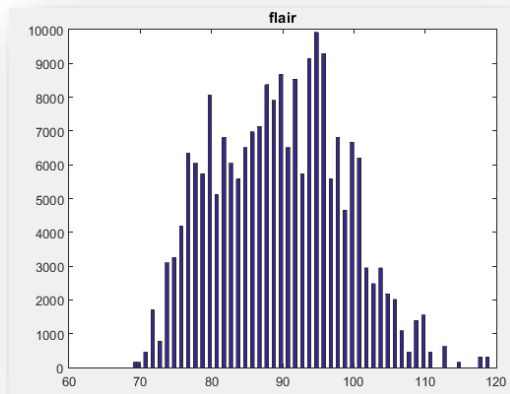
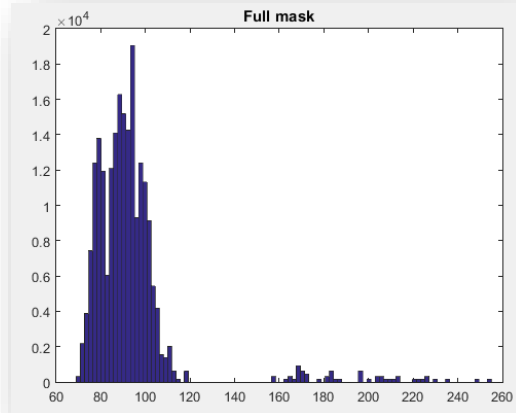
Table2: Shows statistics for patient no.2 individually for each tumor type

Metrics	Tumor1	Tumor2	Tumor4	Overall
Volume	2789	207080	7905	214985
Mean	60.47	191.8039	89.800	93.5515
s.d.	23.99	24.5684	9.07	21.67

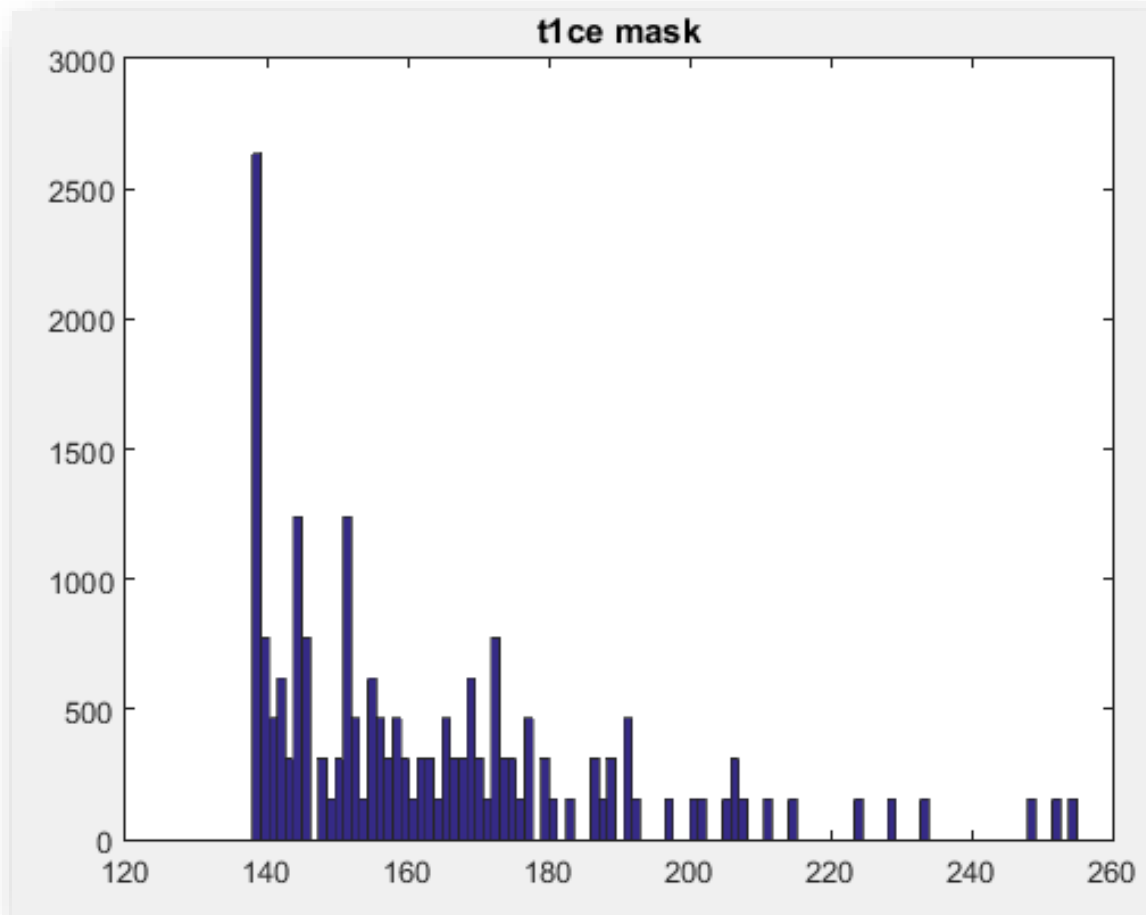
Table3: Shows statistics for patient no.17 individually for each tumor type

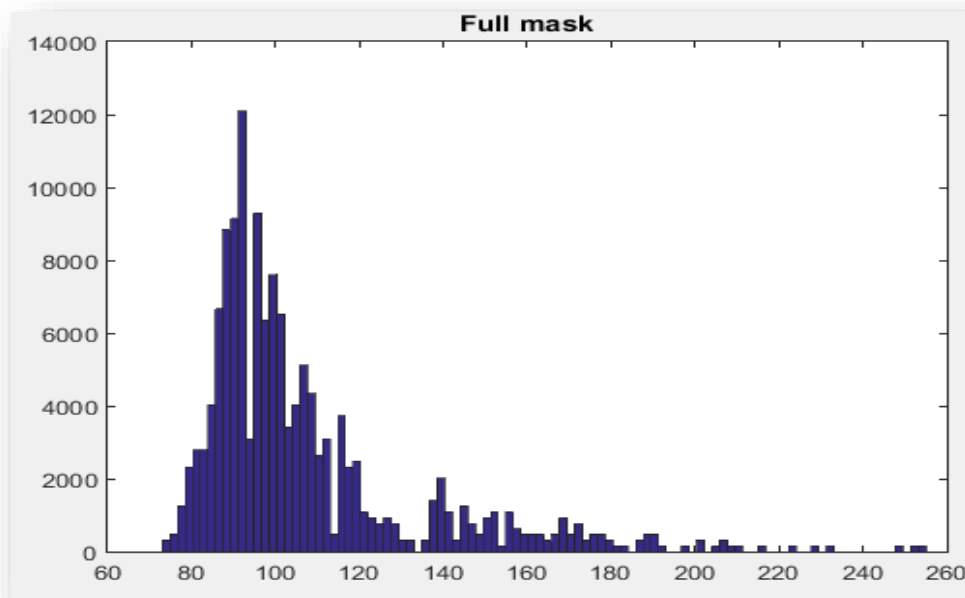
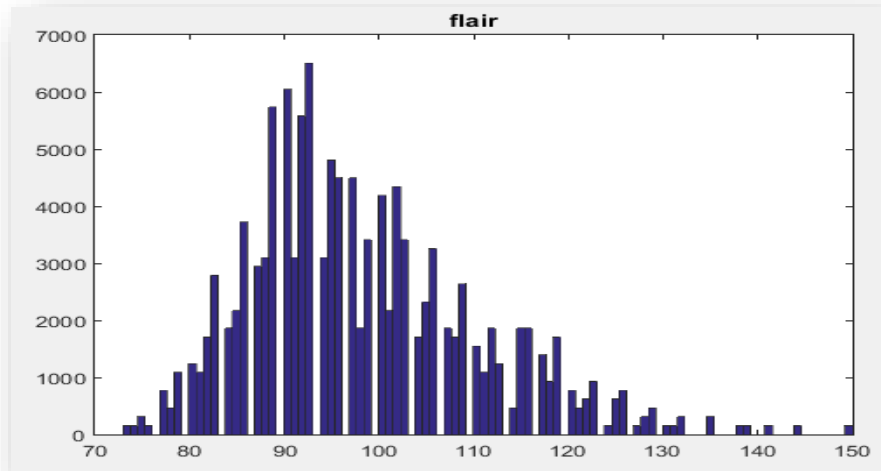
****Histogram Analysis for patient 7 is depicted in previous section itself.**

Patient17: Histograms of tumor tissues.



Patient2: Histograms of tumor tissues.





11. REFERENCES

[1] For Region Growing in 3D

<https://in.mathworks.com/matlabcentral/fileexchange/32532-region-growing-2d-3d-grayscale>

[2] For functions like load_nii

<https://in.mathworks.com/matlabcentral/fileexchange/8797-tools-for-nifti-and-analyze-image>

[3] For K-means used function provided earlier named kmeans1() on moodle.