10/25/2022

Project 3

BME 7450

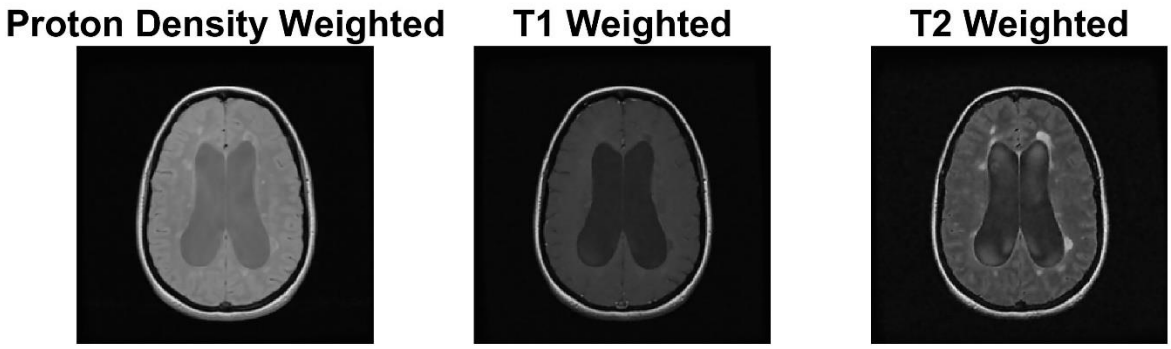
Submitted by,

Rana Mozumder

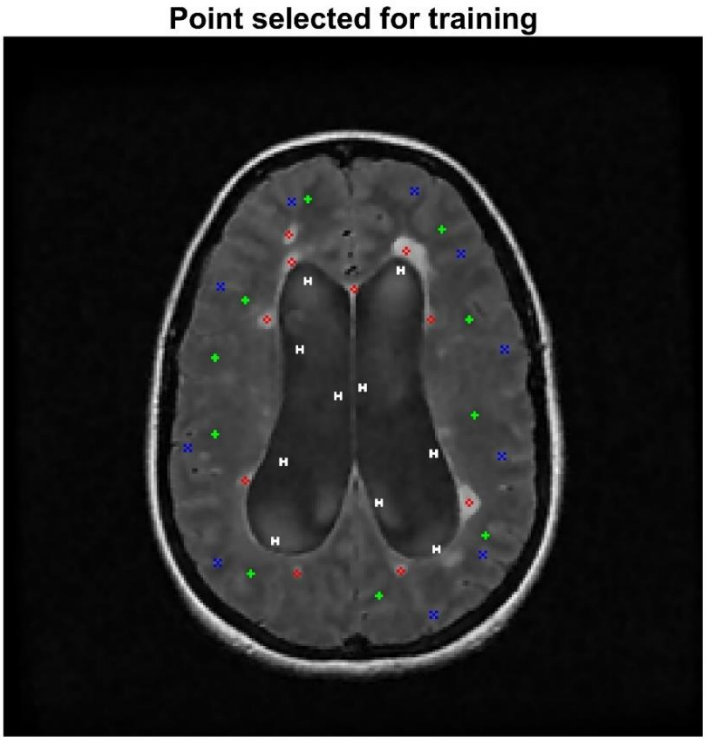
DEPARTMENT OF BME, VANDERBILT UNIVERSITY

VANDERBILT UNIVERSITY

- **All three images of different contrast (PD, T1, and T2 weighted images)**

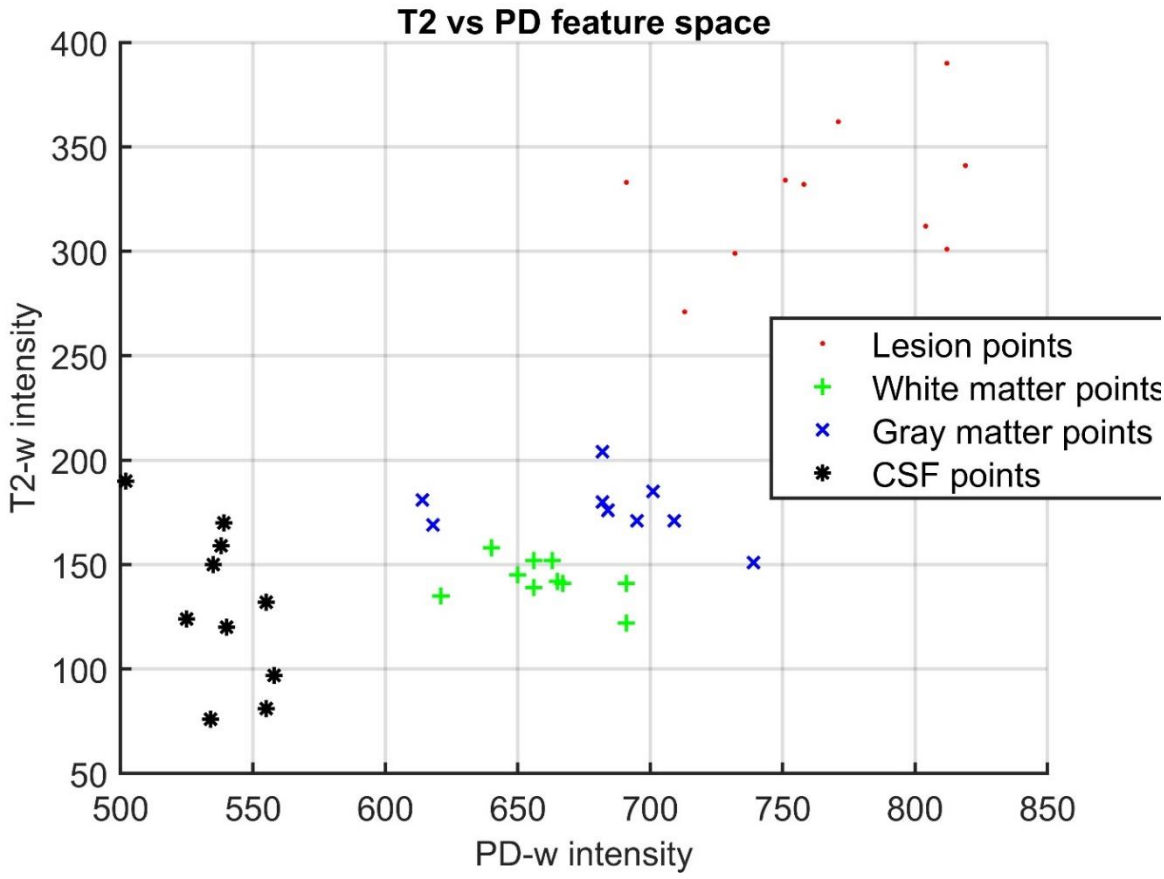


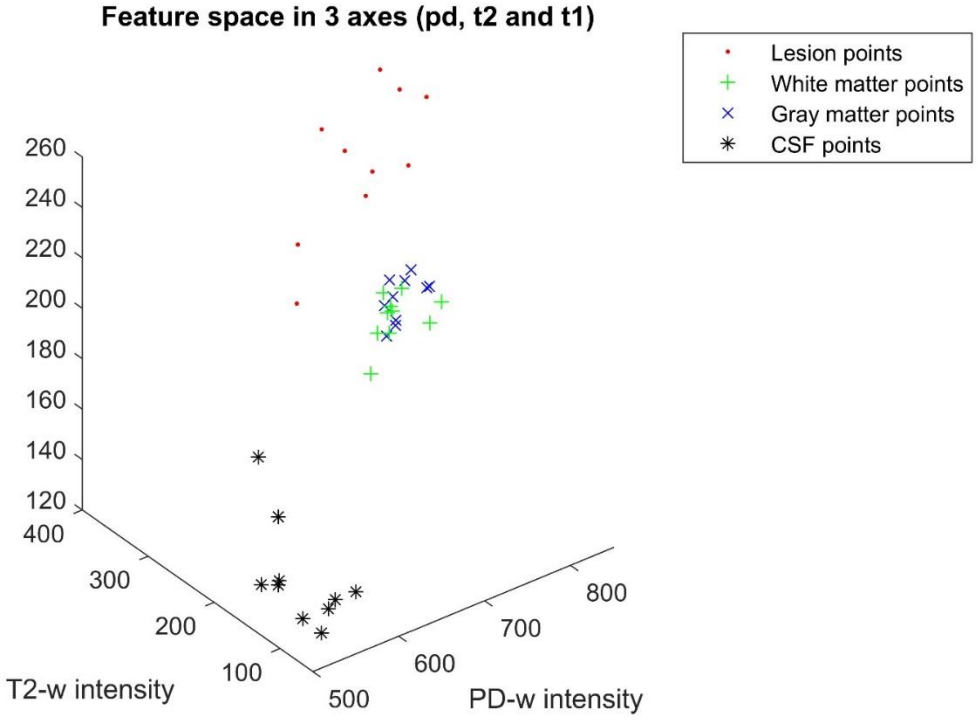
- **T2-weighted image showing the locations of all training points.**



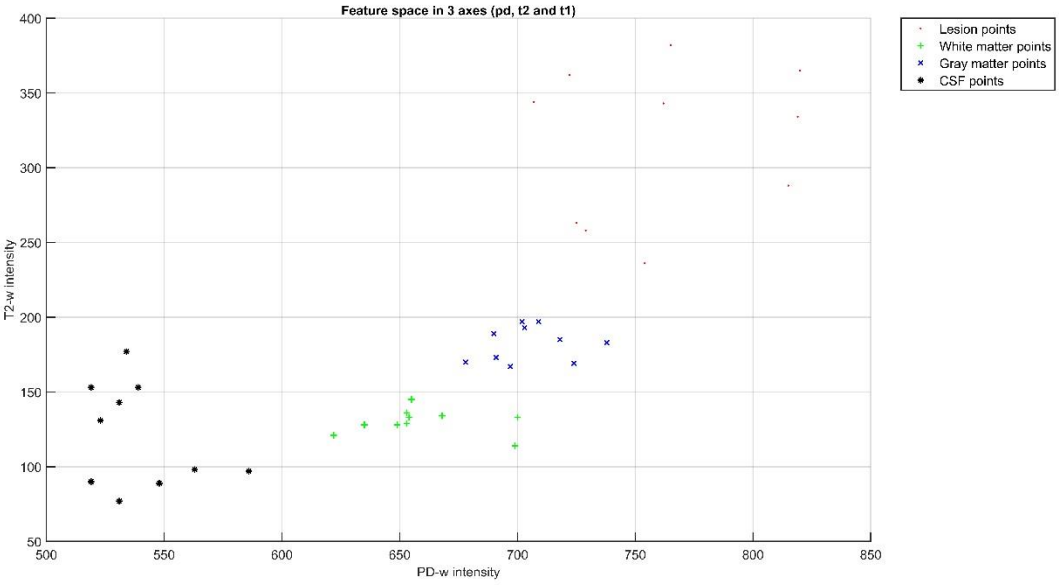
Here,
Red dots are for lesion points,
Green +s are for white matter points,
Blue xs are for gray matter points, and
White Hs are for CSF points.

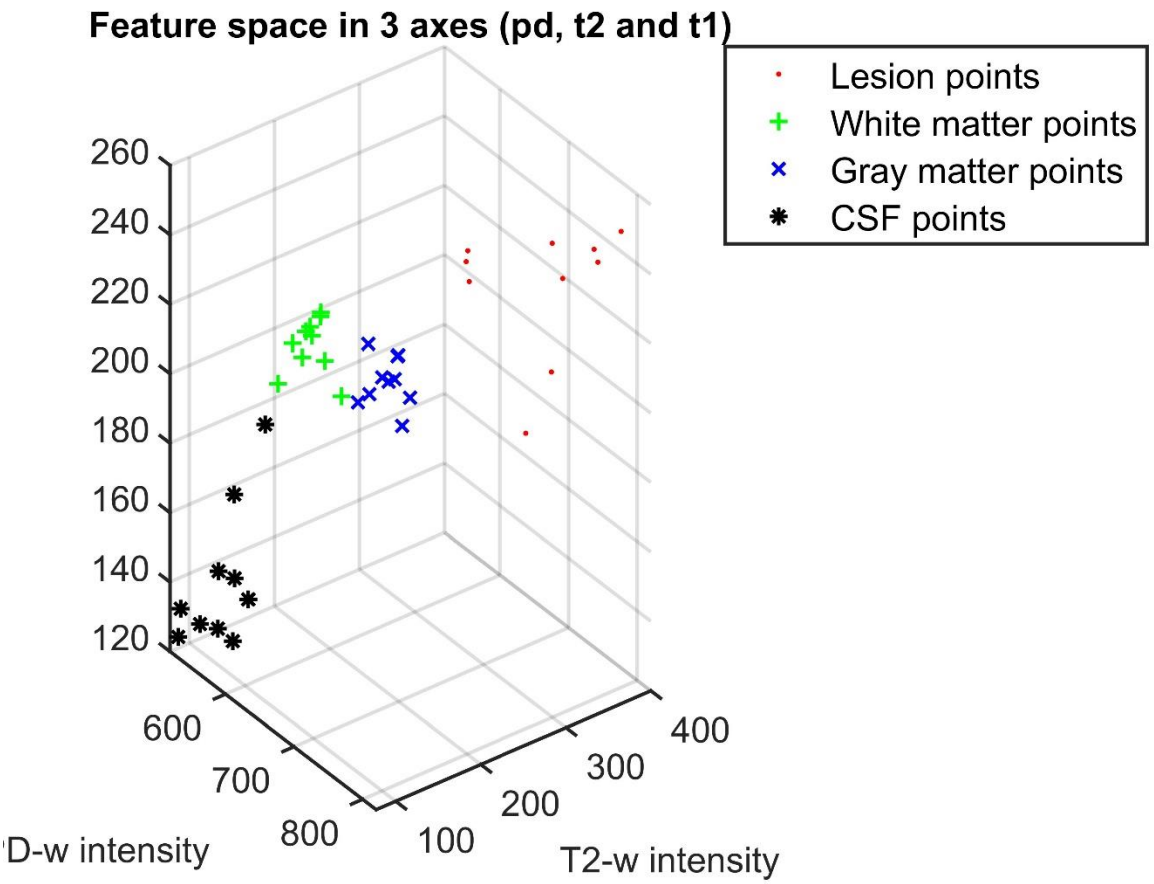
The plot(s) of training point locations in feature space



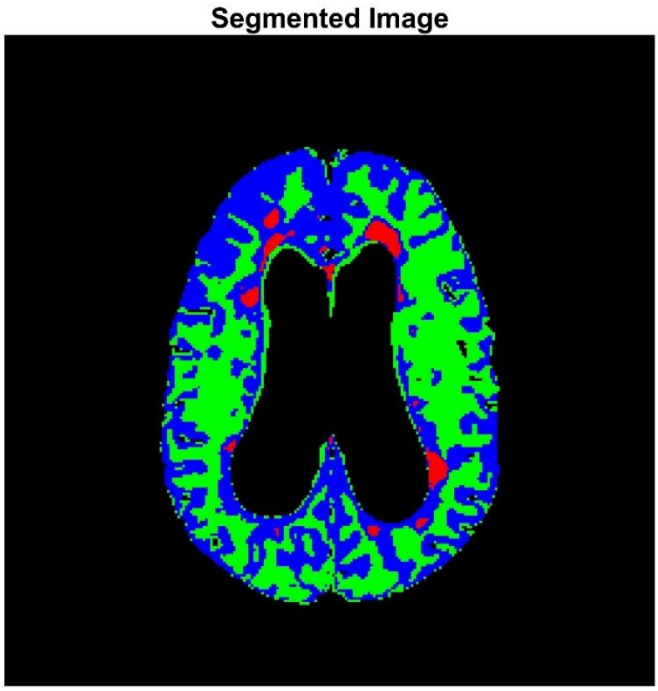


Some more views of this 3D feature space plot.

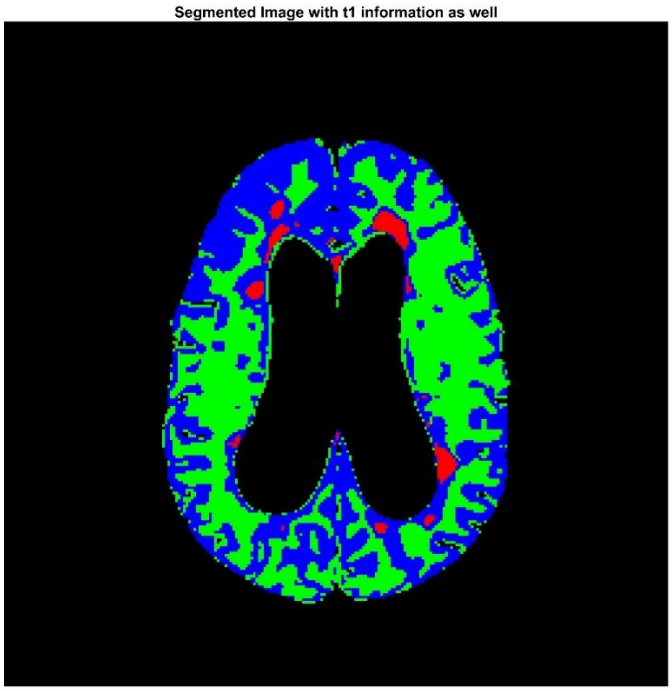




- **The segmentation map with T2 and PD.**



Now, Segmented image using all three different contrast images

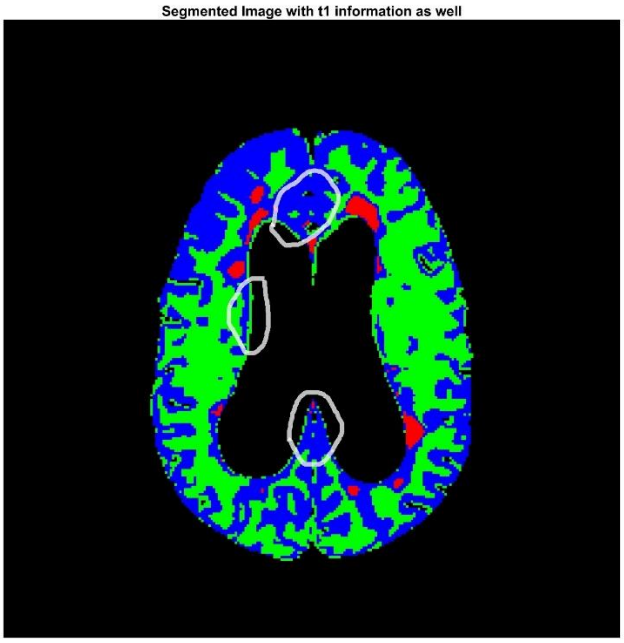


Questions

1. *How accurate is your final segmentation? If there are regions where the tissue classification seems wrong, indicate these on your segmentation map (you can just draw a circle around them, for example).*

   Answer:

   The segmentation of the different tissue region is quite good. But regions are not correctly classified as shown below.



2. *Can you think of any ways to make the segmentation more accurate? What would you change in your algorithm or image acquisition? Can you think of another way to exclude the extracranial fat?*

   Answer:

   Here are some ways to make this segmentation process better:

   - Taking more training points;
   - Using a t2 image with better resolution and contrast;
   - Instead of using Euclidean distance method to categorize pixels, we can use sophisticated clustering algorithms to categorize the pixels.

   To exclude extracranial fat, the following algorithm may work.

   - Since this particular region appears white in all tissue types, if we subtract the t2 weighted image from the PD weighted image, the pixels from this region should become zero.
   - But the subtraction will also lower the intensity value in the pixels that are actually in the brain.
   - Now, if we make a binary mask with the condition that pixel that have intensity greater than zero, we will get a binary mask for the brain region only.

- Now using a for loop, we will only show the value for the pixels in the binary mask that have intensity of 1.

3. *What is the total number of lesion voxels?*
   Answer:
   With t2 and PD information, number of lesion voxel selected was 372. Upon adding t1 information, 432 lesion pixels were categorized.
4. *According to your 3D feature space plot, does T1 weighting provide information not available from T2 weighting? Does it significantly improve the segmentation map?*
   Answer:
   Adding T1 weighting information does make the clusters for all four types of tissue to be distinguishable more easily. So, we can say that T1 weighting provide information not available from T2 and PD weighting. However, after the segmentation plot was created, there were not much improvement in differentiating between tissue areas. There were more accurate classification but not significant.

Matlab Scripts

```
clc; close all;

load('proj3Data.mat');

%selecting training points for lesion region
nTrain = 10;
figure
imagesc(t2w_m)
colormap(gray)
axis image
disp(['Click on’, num2str(nTrain), ‘ lesion points'])
[x_v, y_v] = ginput(nTrain);
row_v = round(y_v);
col_v = round(x_v);

%marking the selected lesion points
L_points = insertMarker(t2w_m/max(t2w_m(:)),[col_v
row_v],'o','color','r', 'Size', 1);

%finding the intensity in PD, t1 and t2 weighted image for the
selected points
index_v = sub2ind(size(t2w_m), row_v, col_v);
t2wl_v = t2w_m(index_v); % T2 weighted intensity for lesion points.
pdwl_v = pdw_m(index_v); % PD weighted intensity for lesion points.
t1wl_v = t1w_m(index_v); % T1 weighted intensity for lesion points.

%mean intensity value for lesion region in PD, t1 and t2 weighted
image
t2wl = mean(t2wl_v);
pdwl = mean(pdwl_v);
t1wl = mean(t1wl_v);

%selecting training points for White Matter region
nTrain = 10;
figure
imagesc(t2w_m)
colormap(gray)
axis image
disp(['Click on’, num2str(nTrain), ‘ White Matter points'])
[x_v, y_v] = ginput(nTrain);
row_v = round(y_v);
col_v = round(x_v);
%marking the selected WM points
```

```
WM_points = insertMarker(L_points,[col_v row_v],'+','color','g',
'Size', 1);

%finding the intensity in PD, t1 and t2 weighted image for the
selected points
index_v = sub2ind(size(t2w_m), row_v, col_v);
t2ww_v = t2w_m(index_v); % T2 weighted intensity for WM points.
pdww_v = pdw_m(index_v); % PD weighted intensity for Wm points.
t1ww_v = t1w_m(index_v); % T1 weighted intensity for Wm points.

%mean intensity value for lesion region in PD,t1 and t2 weighted
image
t2ww = mean(t2ww_v);
pdww = mean(pdww_v);
t1ww = mean(t1ww_v);

%displaying the selected training points for Gray Matter region
nTrain = 10;
figure
imagesc(t2w_m)
colormap(gray)
axis image
axis off
disp(['Click on', num2str(nTrain), ' gray points'])
[x_v, y_v] = ginput(nTrain);
row_v = round(y_v);
col_v = round(x_v);
%marking the selected GM points
GM_points = insertMarker(WM_points,[col_v row_v],...
    'x','color','blue', 'Size', 1);

%finding the intensity in PD, t1 and t2 weighted image for the
selected points
index_v = sub2ind(size(t2w_m), row_v, col_v);
t2wg_v = t2w_m(index_v); % T2 weighted intensity for gm points.
pdwg_v = pdw_m(index_v); % PD weighted intensity for gm points.
t1wg_v = t1w_m(index_v); % T1 weighted intensity for gm points.

%mean intensity value for lesion region in PD,t1 and t2 weighted
image
t2wg = mean(t2wg_v);
pdwg = mean(pdwg_v);
t1wg = mean(t1wg_v);

%displaying the selected training points for CSF region
nTrain = 10;
```

```
figure
imagesc(t2w_m)
colormap(gray)
axis image
disp(['Click on', num2str(nTrain), ' CSF points'])
[x_v, y_v] = ginput(nTrain);
row_v = round(y_v);
col_v = round(x_v);

%marking the selected CSF points
CSF_points = insertMarker(GM_points,[col_v row_v],...
    'star','color','white', 'Size', 1);

%displaying all selected points
figure
image(CSF_points)
axis image
axis off
title('Point selected for training')

%finding the intensity in PD, t1 and t2 weighted image for the
selected points
index_v = sub2ind(size(t2w_m), row_v, col_v);
t2wc_v = t2w_m(index_v); % T2 weighted intensity for csf points.
pdwc_v = pdw_m(index_v); % PD weighted intensity for csf points.
t1wc_v = t1w_m(index_v); % T1 weighted intensity for csf points.

%mean intensity value for lesion region in PD,t1 and t2 weighted
image
t2wc = mean(t2wc_v);
pdwc = mean(pdwc_v);
t1wc = mean(t1wc_v);

%plotting feature space
figure
plot3(pdwl_v, t2wl_v, t1wl_v, 'r.', pdww_v, t2ww_v, t1ww_v, 'g+',
...
    pdwg_v, t2wg_v, t1wg_v, 'bx', pdwc_v, t2wc_v, t1wc_v, 'k*')
xlabel('PD-w intensity')
ylabel('T2-w intensity')

%creating a binary mask
pdMax = max(pdw_m(:));
mask_m = (pdw_m > 0.1*pdMax);
index2_v = find(mask_m(:));
t2wHead_v = t2w_m(index2_v);
```

```
pdwHead_v = pdw_m(index2_v);
t1wHead_v = t1w_m(index2_v);
nHeadPixels = length(index2_v);

lesionMask_m = zeros(256, 256);
wmMask_m = zeros(256, 256);
gmMask_m = zeros(256, 256);
csfMask_m = zeros(256, 256);
for pixel = 1:nHeadPixels

    t2w = t2wHead_v(pixel);
    pdw = pdwHead_v(pixel);
    t1w = t1wHead_v(pixel);

    lesion_dist = sqrt((t2w-t2wl)^2 + (pdw-pdwl)^2 + (t1w-t1wl)^2);
    wm_dist = sqrt((t2w-t2ww)^2 + (pdw-pdww)^2 + (t1w-t1ww)^2);
    gm_dist = sqrt((t2w-t2wg)^2 + (pdw-pdwg)^2 + (t1w-t1wg)^2);
    csf_dist = sqrt((t2w-t2wc)^2 + (pdw-pdwc)^2 + (t1w-t1wc)^2);

    type = min([lesion_dist, wm_dist, gm_dist, csf_dist]);
    if type == lesion_dist
        lesionMask_m(ind2sub(size(t2w_m), index2_v(pixel))) = 1;
    elseif type == wm_dist
        wmMask_m(ind2sub(size(t2w_m), index2_v(pixel))) = 1;
    elseif type == gm_dist
        gmMask_m(ind2sub(size(t2w_m), index2_v(pixel))) = 1;
    else
        csfMask_m(ind2sub(size(t2w_m), index2_v(pixel))) = 1;
    end
end

%displaying segmented image with extracranial fat
figure
image(cat(3, lesionMask_m, wmMask_m, gmMask_m))
axis image
axis off

%removing the fat
disp('Define a polygon enclosing brain, excluding extracranial fat')
skullMask_m = roipoly(pdw_m / pdMax);
%Zero all pixels outside the skull:
lesionMask_m = skullMask_m .* lesionMask_m;
wmMask_m = skullMask_m .* wmMask_m;
gmMask_m = skullMask_m .* gmMask_m;
%Display improved segmentation map:
figure
```

```
image(cat(3, lesionMask_m, wmMask_m, gmMask_m))
axis image
axis off

%calculating number of pixels for all tissues
n_lesion_pix = sum(lesionMask_m(:))
n_wm_pix = sum(wmMask_m(:))
n_gm_pix = sum(gmMask_m(:))
n_csf_pix = sum(csfMask_m(:))
```