12/1/2022

Project 7

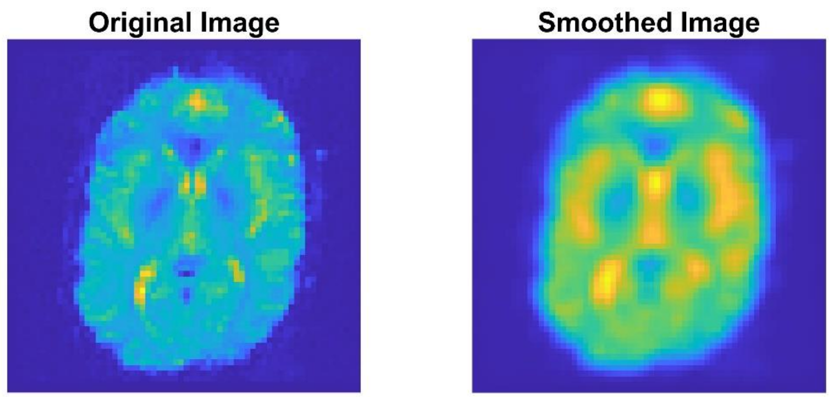
BME 7450

Submitted by,

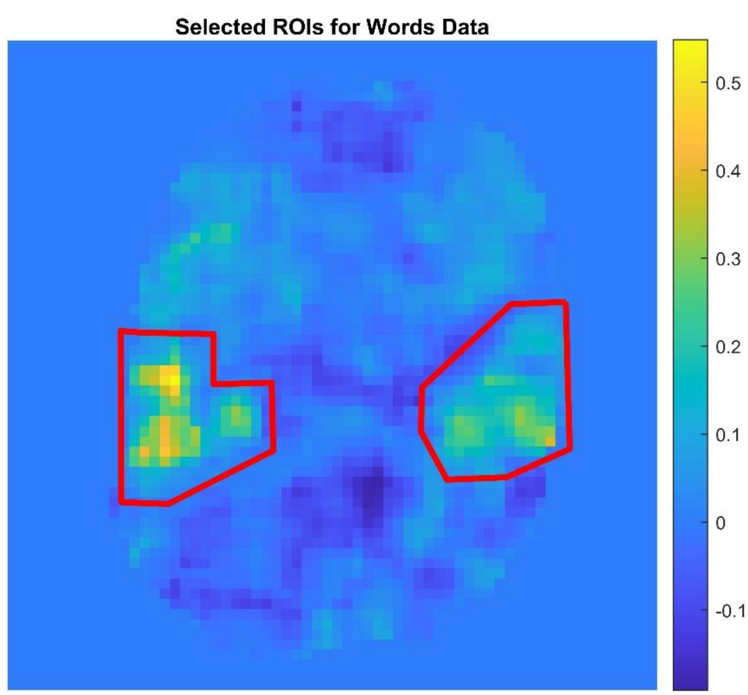
Rana Mozumder

DEPARTMENT OF BME, VANDERBILT UNIVERSITY

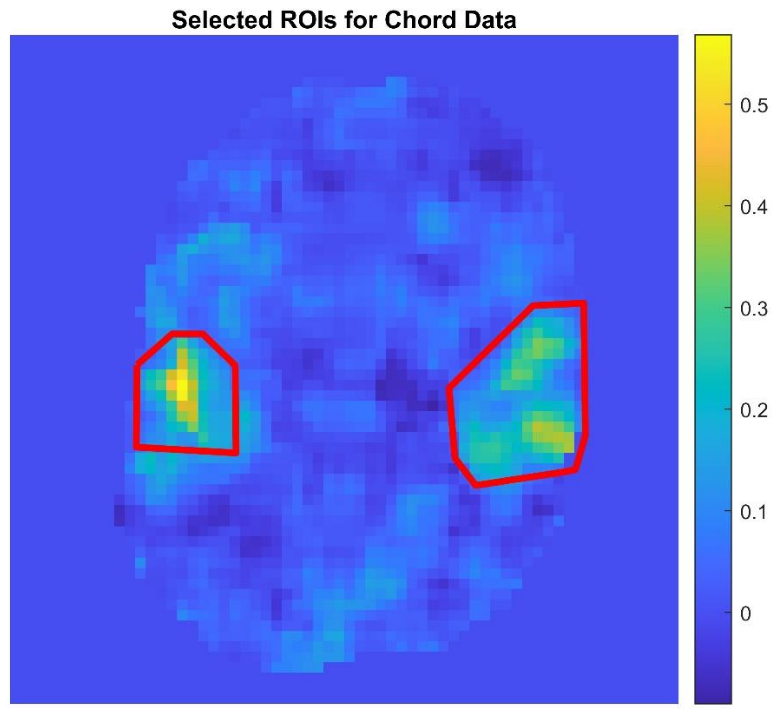
An original vs smoothed image



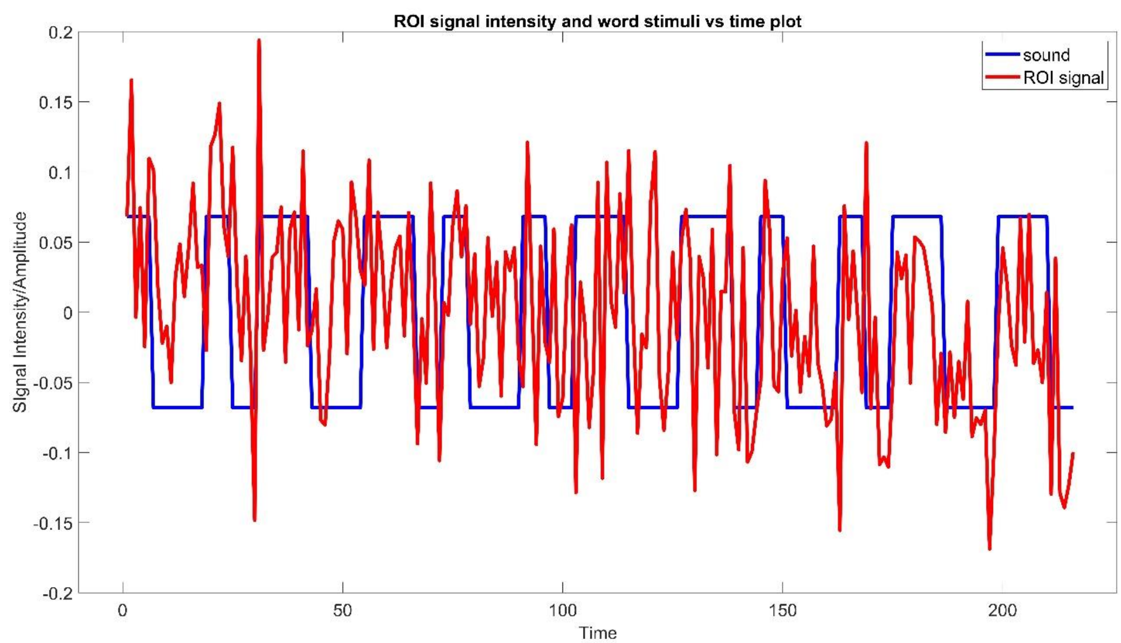
‘Word’ correlation coefficient map with selected ROIs



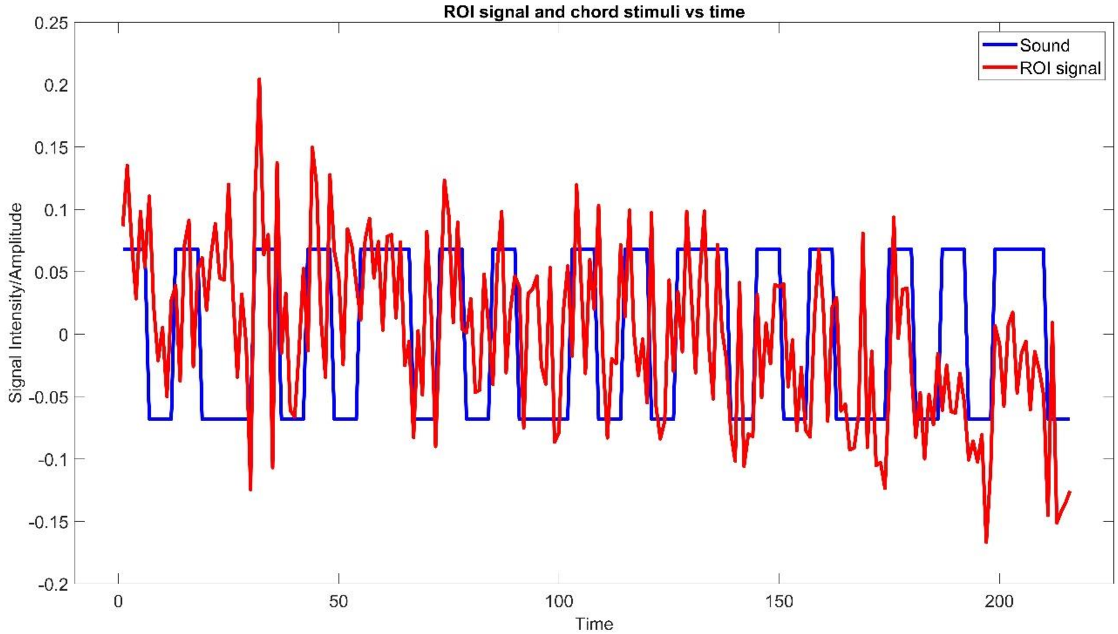
'Chord' correlation coefficient map with selected ROIs



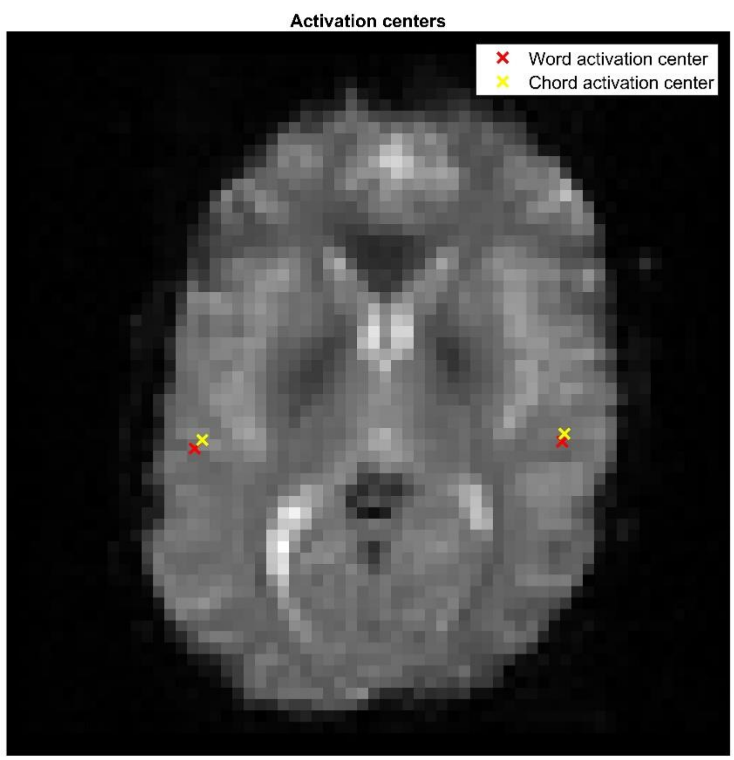
ROI signal intensity vs time for 'Word' stimuli



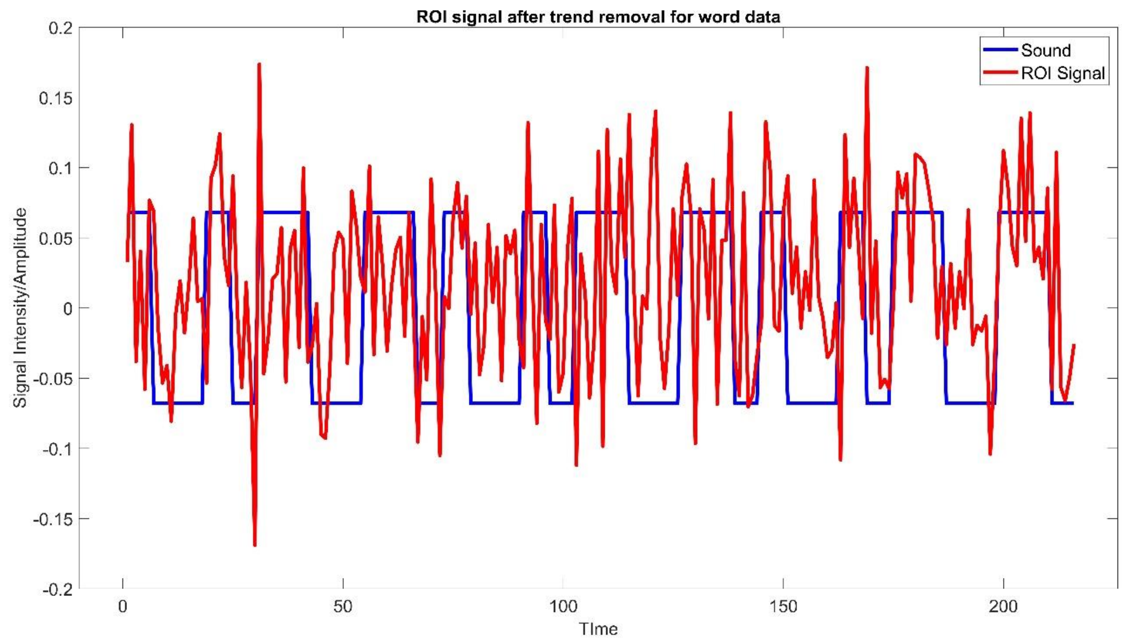
ROI signal intensity vs time for 'Chord' stimuli



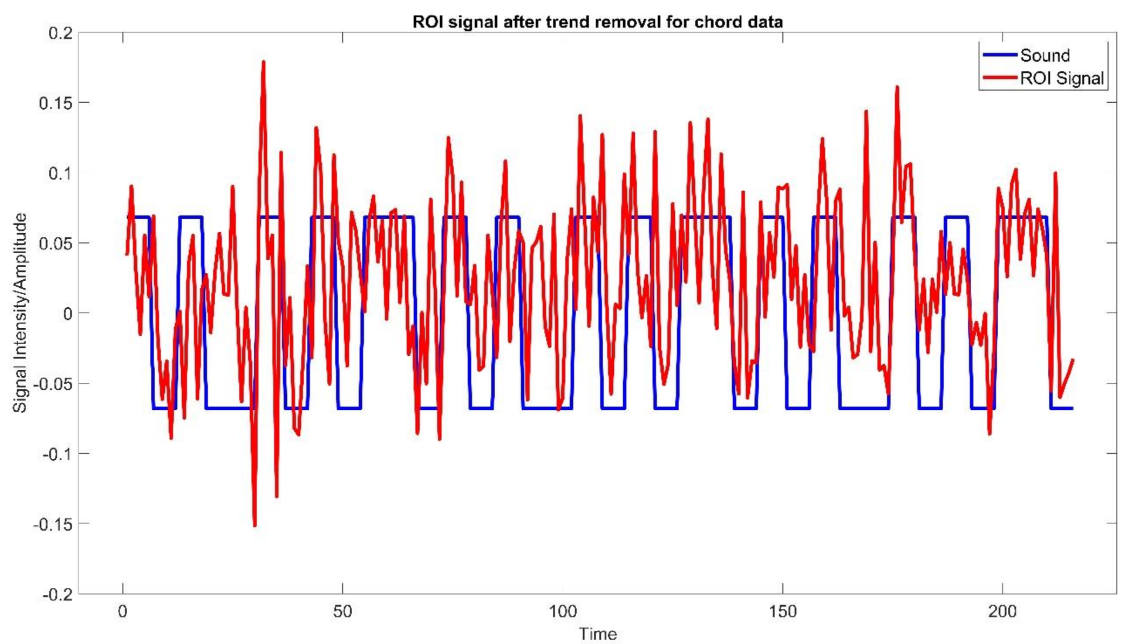
A figure showing the location of the center of Word and Chord activations in each hemisphere



Plot of ROI signal intensity versus time after trend removal for 'Word' data



Plot of ROI signal intensity versus time after trend removal for 'Chord' data



Questions

1. *Why does the signal increase when the subject hears the sound (words or chords)?*

   Answer:

   When the subjects hear the sound certain neurons in the brain starts firing which needs influx of oxygen. So, when neurons start to fire, there is an influx of oxygenated blood in the brain. Hence, Blood oxygen level dependent (BOLD) signal increases.

2. *What is the contrast-to-noise ratio of the BOLD measurement for each stimulus? What are some strategies (beside step #12) that you could use to improve this?*

   Answer:

   CNR for word data = 0.4377

   CNR for chord data = 0.5448

   To improve CNR, we can use the following strategies:

   I. We can smooth the signal separately when stimulus is on and off. Thus, difference between the means of stimuli on and off would be bigger. Also, standard deviations would decrease. This is eventually increase CNR.
   II. Instead of fitting a straight line, fitting a non-linear (exponential) line would be better since the drift was non-linear in nature.

3. *Studies have shown that speech, like all sounds, is processed by the 'primary' auditory cortex, but unlike other stimuli, the information is then fed to a neighboring area farther back (i.e., posterior) in the temporal lobe. Are your measurements of the centers of activation consistent with this model? If the primary auditory area detects the frequency and timing of sounds, what do you think the neighboring area might do? The 'words' used in our experiment are comprised of standard English phonemes (i.e., sounds, for example 'vos'), but have no meaning.*

   Answer:

   Measurements show that there were multiple spots that got activated by the word and chord stimuli which is in line with the model.

   While the primary area detects frequency, I think the other areas might measure the source of sound or meaningfulness of the sound.

4. *What activation CNR did you measure for each stimulus after removing the linear trend in signal? Did this step improve the CNR? Why does your result make sense?*

   Answer:

   After removing the linear trend,

   CNR for word data = 0.4666

   CNR for chord data = 0.5868

   Yes, removal of trend improved CNR. Due change in scanner sensitivity, we can see a drift in the recorded signal. So, when we corrected it, CNR was bound to increase. This is also because due to the drift the mean and standard deviation calculation were not accurate which reduced the CNR. Removal of the drift also helps in this aspect.

Matlab code

```
clc; close all; clear all;

%loading the word data
load('proj7wordsData.mat');

%smoothing the images
nTimes = length(sound_v);
for time = 1:nTimes
    image_m = squeeze(image_3d(:, :, time));
    smoothImage_m = conv2(image_m, ones(4), 'same');
    image2_3d(:, :, time) = smoothImage_m;
    % displaying both the original and smoothed images (in one figure):
    if time ==1
        figure
        subplot 121
        imagesc(image_m)
        axis image
        axis off
        title('Original Image')

        subplot 122
        imagesc(smoothImage_m)
        axis image
        axis off
        title('Smoothed Image')
    end
end

%creating head mask
headMask_m = zeros(64, 64);
max_pixel = max(image2_3d(:));
for row=1:64
    for col=1:64
        pixels = squeeze(image2_3d(row, col, :));
        if mean(pixels)>=0.1*max_pixel
            headMask_m(row, col) =1;
        end
    end
end
%displaying the head mask
figure
imagesc(headMask_m)
colormap(gray)
axis image
axis off
title("Head Mask")

%normalizing the sound vector
normSound_v = (sound_v-mean(sound_v))/norm(sound_v-mean(sound_v));

%creating correlation coefficient map for words
rWord_m = zeros(64, 64);
for row=1:64
    for col=1:64
        if headMask_m(row, col) ==1
            data_v = squeeze(image2_3d(row, col, :));
```

```
            % Insert code here to transform the data_v to have zero mean
            % and unit norm. Call this vector normData_v:
            normData_v = (data_v-mean(data_v))/norm(data_v-mean(data_v));
            % Calculate the correlation coefficient of normData_v and
            % normSound_v:
            rWord_m(row, col) = sum(normData_v .* normSound_v);
        end
    end
end
figure
imagesc(rWord_m)
axis image
axis off
colorbar
title("Correlation Coefficients Map for words")


title('Define left hemisphere region of interest...')
[leftRoiMask_m, xLeft_v, yLeft_v] = roipoly;
line(xLeft_v, yLeft_v, 'LineWidth', 3, 'Color', 'r')
title('Define right hemisphere region of interest...')
[rightRoiMask_m, xRight_v, yRight_v] = roipoly;
line(xRight_v, yRight_v, 'LineWidth', 3, 'Color', 'r')
roiMask_m = rightRoiMask_m + leftRoiMask_m;
roiSignal_v = zeros(nTimes, 1);

%calculating roiSignal_v at each time point:
for time = 1:nTimes
    roiSignal_v(time,1) = sum(roiMask_m.*squeeze(image2_3d(:,:,time)), 'all');
end

%normalizing the ROI signal
normRoiSignal_v = (roiSignal_v-mean(roiSignal_v))/norm(roiSignal_v-mean(roiSignal_v));

%plotting the signal and sound against time
time_v = 1:length(sound_v);
figure
plot(time_v, normSound_v', 'b', time_v, normRoiSignal_v', 'r', 'LineWidth', 3)

%separating signal during stimuli on and off
stim = 1;
rest = 1;
for time=1:nTimes
    if sound_v(time, 1)==1
        sigOn(1, stim) = normRoiSignal_v(time,1);
        stimTime(1, stim) = time;
        stim = stim+1;
    else
        sigOff(1, rest) = normRoiSignal_v(time,1);
        restTime(1, rest) = time;
        rest = rest+1;
    end
end

%calculating means
meanStim = sum(sigOn)/sum(sound_v);
meanRest = sum(sigOff)/(216-sum(sound_v));

%calculating standard deviations
stdStim = std(sigOn);
stdRest = std(sigOff);
```

```
%%CNR calculation

CNR_w = (meanStim-meanRest)/(sqrt(stdStim^2 + stdRest^2));

%calculating the center for word
[cols_m, rows_m] = meshgrid(1:64, transpose(1:64));
wordLeftCenterRow = sum(sum(leftRoiMask_m.*rows_m.*rWord_m)) / ...
    sum(sum(leftRoiMask_m .* rWord_m));
wordLeftCenterCol = sum(sum(leftRoiMask_m.*cols_m.*rWord_m)) / ...
    sum(sum(leftRoiMask_m .* rWord_m));


wordRightCenterRow = sum(sum(rightRoiMask_m.*rows_m.*rWord_m)) / ...
    sum(sum(rightRoiMask_m .* rWord_m));
wordRightCenterCol = sum(sum(rightRoiMask_m.*cols_m.*rWord_m)) / ...
    sum(sum(rightRoiMask_m .* rWord_m));




%eliminating the linear trend
p = polyfit(restTime, sigOff, 1);
y = polyval(p, time_v);
corrected_normRoiSignal_v = normRoiSignal_v'-y;
figure
plot(time_v, normSound_v', 'b', time_v, corrected_normRoiSignal_v, 'r', 'LineWidth', 3)
hold on
plot(time_v, y)

%%% for corrected CNR calculation
%separating signal during stimuli on and off
stim = 1;
rest = 1;
for time=1:nTimes
    if sound_v(time, 1)==1
        correctedsigOn(1, stim) = corrected_normRoiSignal_v(1, time);
        stim = stim+1;
    else
        correctedsigOff(1, rest) = corrected_normRoiSignal_v(1, time);
        rest = rest+1;
    end
end

%calculating means
correctedmeanStim = sum(correctedsigOn)/sum(sound_v);
correctedmeanRest = sum(correctedsigOff)/(216-sum(sound_v));

%calculating standard deviations
correctedstdStim = std(correctedsigOn);
correctedstdRest = std(correctedsigOff);

CNR_w_corrected = (correctedmeanStim-correctedmeanRest)/(sqrt(correctedstdStim^2 +
correctedstdRest^2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%loading chord data
load('proj7chordsData.mat');
```

```
%image smoothing
nTimes = length(sound_v);
for time = 1:nTimes
    image_m = squeeze(image_3d(:, :, time));
    smoothImage_m = conv2(image_m, ones(4), 'same');
    image2_3d(:, :, time) = smoothImage_m;

    if time ==1
        figure
        subplot 121
        imagesc(image_m)
        axis image
        axis off
        title('Original Image')

        subplot 122
        imagesc(smoothImage_m)
        axis image
        axis off
        title('Smoothed Image')
    end
end

%headmask
headMask_m = zeros(64, 64);
max_pixel = max(image2_3d(:));
for row=1:64
    for col=1:64
        pixels = squeeze(image2_3d(row, col, :));
        if mean(pixels)>=0.1*max_pixel
            headMask_m(row, col) =1;
        end
    end
end
%displaying the head mask
figure
imagesc(headMask_m)
colormap(gray)
axis image
axis off
title("Head Mask")

%normalizing chord sound
normSound_c_v = (sound_v-mean(sound_v))/norm(sound_v-mean(sound_v));

%chord correlation coefficient map
rChords_m = zeros(64, 64);
for row=1:64
    for col=1:64
        if headMask_m(row, col) ==1
            data_v = squeeze(image2_3d(row, col, :));
            % Insert code here to transform the data_v to have zero mean
            % and unit norm. Call this vector normData_v:
            normData_v = (data_v-mean(data_v))/norm(data_v-mean(data_v));
            % Calculate the correlation coefficient of normData_v and
            % normSound_v:
            rChords_m(row, col) = sum(normData_v .* normSound_c_v);
        end
    end
end
figure
```

```
imagesc(rChords_m)
axis image
axis off
colorbar
title("Correlation Coefficients Map for Chords")


title('Define left hemisphere region of interest...')
[leftRoiMask_c_m, xLeft_v, yLeft_v] = roipoly;
line(xLeft_v, yLeft_v, 'LineWidth', 3, 'Color', 'r')
title('Define right hemisphere region of interest...')
[rightRoiMask_c_m, xRight_v, yRight_v] = roipoly;
line(xRight_v, yRight_v, 'LineWidth', 3, 'Color', 'r')
roiMask_c_m = rightRoiMask_c_m + leftRoiMask_c_m;
roiSignal_c_v = zeros(nTimes, 1);

% calculating roiSignal_v at each
% time point:
for time = 1:nTimes
    roiSignal_c_v(time,1) = sum(roiMask_c_m.*squeeze(image2_3d(:,:,time)), 'all');
end

%normalizing signal
normRoiSignal_c_v = (roiSignal_c_v-mean(roiSignal_c_v))/norm(roiSignal_c_v-
mean(roiSignal_c_v));

%plotting signal
time_v = 1:length(sound_v);
figure
plot(time_v, normSound_c_v', 'b', time_v, normRoiSignal_c_v', 'r', 'LineWidth', 3)

%for CNR calculation
stim_c = 1;
rest_c = 1;
for time=1:nTimes
    if sound_v(time, 1)==1
        sigOn_c(1, stim_c) = normRoiSignal_c_v(time,1);
        stimTime_c(1, stim_c) = time;
        stim_c = stim_c+1;
    else
        sigOff_c(1, rest_c) = normRoiSignal_c_v(time,1);
        restTime_c(1, rest_c) = time;
        rest_c = rest_c+1;
    end
end

meanStim_c = sum(sigOn_c)/sum(sound_v);
meanRest_c = sum(sigOff_c)/(216-sum(sound_v));

stdStim_c = std(sigOn_c);
stdRest_c = std(sigOff_c);

%%CNR calculation
CNR_c = (meanStim_c-meanRest_c)/(sqrt(stdStim_c^2 + stdRest_c^2));

%%%calculating the source
[cols_m, rows_m] = meshgrid(1:64, transpose(1:64));
chordLeftCenterRow = sum(sum(leftRoiMask_c_m.*rows_m.*rChords_m)) / ...
    sum(sum(leftRoiMask_c_m .* rChords_m));
chordLeftCenterCol = sum(sum(leftRoiMask_c_m.*cols_m.*rChords_m)) / ...
    sum(sum(leftRoiMask_c_m .* rChords_m));
```

```
chordRightCenterRow = sum(sum(rightRoiMask_c_m.*rows_m.*rChords_m)) / ...
    sum(sum(rightRoiMask_c_m .* rChords_m));
chordRightCenterCol = sum(sum(rightRoiMask_c_m.*cols_m.*rChords_m)) / ...
    sum(sum(rightRoiMask_c_m .* rChords_m));


%eliminating the linear trend
p = polyfit(restTime_c, sigOff_c, 1);
y = polyval(p, time_v);
corrected_normRoiSignal_c_v = normRoiSignal_c_v'-y;
figure
plot(time_v, normSound_c_v', 'b', time_v, corrected_normRoiSignal_c_v, 'r',
'LineWidth', 3)



%separating signal during stimuli on and off
stim = 1;
rest = 1;
for time=1:nTimes
    if sound_v(time, 1)==1
        correctedsigOn(1, stim) = corrected_normRoiSignal_c_v(1, time);
        stim = stim+1;
    else
        correctedsigOff(1, rest) = corrected_normRoiSignal_c_v(1, time);
        rest = rest+1;
    end
end

%calculating means
correctedmeanStim = sum(correctedsigOn)/sum(sound_v);
correctedmeanRest = sum(correctedsigOff)/(216-sum(sound_v));

%calculating standard deviations
correctedstdStim = std(correctedsigOn);
correctedstdRest = std(correctedsigOff);

CNR_c_corrected = (correctedmeanStim-correctedmeanRest)/(sqrt(correctedstdStim^2 +
correctedstdRest^2));




%%%%%%%%%%%%%%%%
%source location plot

figure
imagesc(squeeze(image_m(:,:,1)))
axis image
axis off
colormap(gray)
title("Activation centers")
hold on
plot(wordLeftCenterCol, wordLeftCenterRow, 'rx', 'MarkerSize', 10, 'LineWidth', 2)
hold on
plot(wordRightCenterCol, wordRightCenterRow, 'rx', 'MarkerSize', 10, 'LineWidth', 2)
hold on
plot(chordLeftCenterCol, chordLeftCenterRow, 'yx', 'MarkerSize', 10, 'LineWidth', 2)
```

```
hold on
plot(chordRightCenterCol, chordRightCenterRow, 'yx', 'MarkerSize', 10, 'LineWidth', 2)
legend('Word activation center','', 'Chord activation center')
hold off
```