**Lab 3:**

**Q1:** Construct the dictionary basis matrix as I provided you the code in MATLAB and also check the KSVD dictionary in KSVD toolbox.

Construct dictionary with different size

say **8x8 (64x1024), 10x10 (100x2014), 16x16 (256x1024)**

**save all dictionaries in matfile**

**see function for how to construct dictionary**

**basis={};**

**M=8; N=8;**

**M1=M^2; N1=N^2;**

**M2=M; N2=M;**

**count1=0;**

**for k1=1:0.5:M1**

**if k1==1; nf1=sqrt(1/M); else nf1=sqrt(2/M); end**

**for k2=1:0.5:N1**

**count1=count1+1;**

**if k2==1; nf2=sqrt(1/N); else nf2=sqrt(2/N); end**

**for n1=1:M2**

**for n2=1:N2**

**temp(n1,n2)=nf1\*nf2\*cos(( (2\*(n1-1)+1)\*(k1-1)\*pi)/(2\*M)) \* cos(( (2\*(n2-1)+1)\*(k2-1)\*pi)/(2\*N));**

**end**

**end**

**Dl(:,count1)=temp(:);**

**end**

**end**

**xp=randperm(size(Dl,2));**

**Dl=Dl(:,xp);**

**DicDorm = sqrt(sum(Dl.^2));**

**lNorm = sqrt(sum(Dl.^2));**

**Idx = find(lNorm);**

**Dl = Dl(:, Idx);**

**% % Dl = Dl./repmat(sqrt(sum(Dl.^2)), size(Dl, 1), 1);**

**dict\_size = 1024; % dictionary size**

**Dl=Dl(:,1:dict\_size );**

**basis=Dl;**

**count=1;**

**d=zeros(128,128);**

**for n1=1:16**

**lx=(n1-1)\*N+1; hx=lx+N-1;**

**for n2=1:16**

**ly=(n2-1)\*N+1; hy=ly+N-1;**

**d(lx:hx, ly:hy)=reshape(basis(:, count),N,N);**

**% if n1==8**

**% n1**

**% reshape(basis(:, count),8,8)**

**% end**

**count=count+1;**

**if count > 256**

**break**

**end**

**end**

**if count > 256**

**break**

**end**

**end**

**imshow(abs(d),[])**

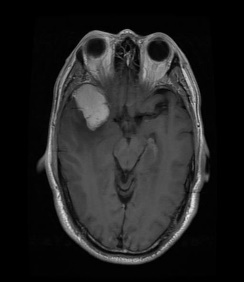
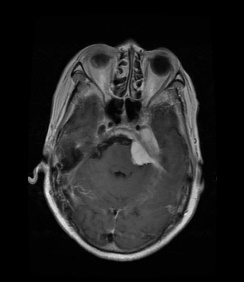
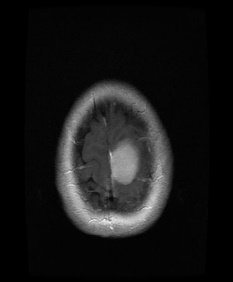
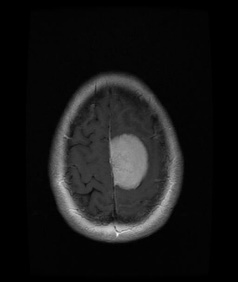
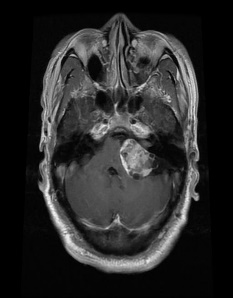
**Q2:** Construct the feature matrix based on dictionaries (already constructed in Q1) using OMP (orthogonal matching pursuit) optimization algorithm using

1) Image classification dataset (brain tumor dataset). The dataset samples are given below.

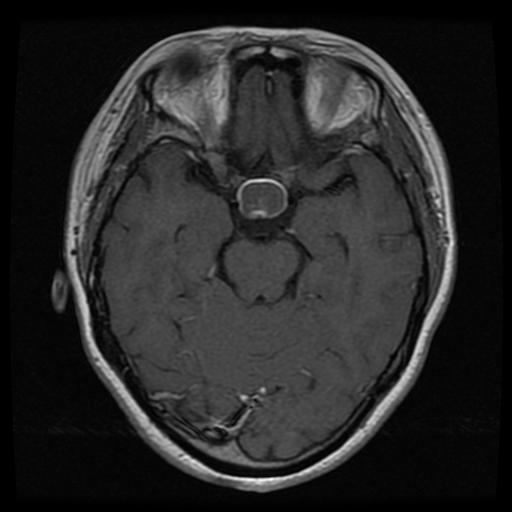
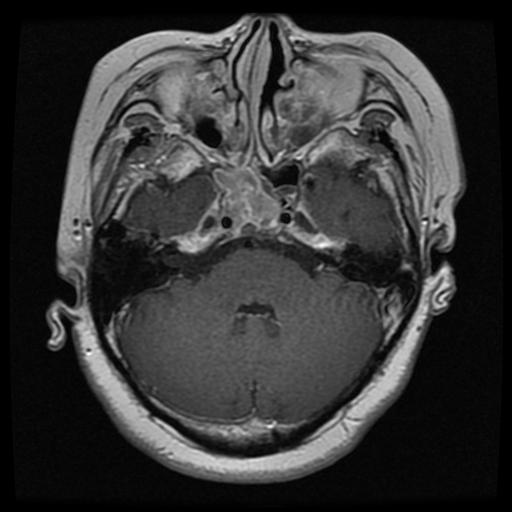
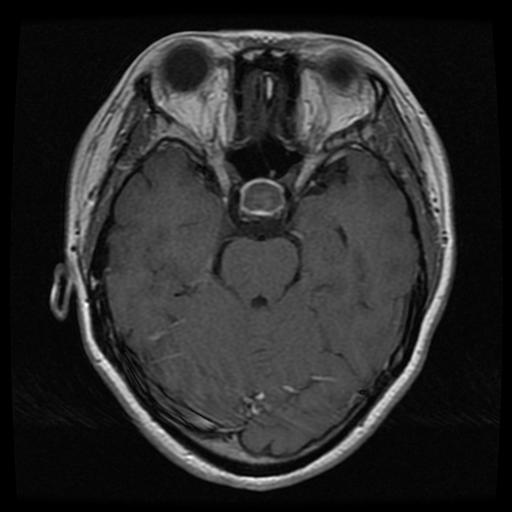
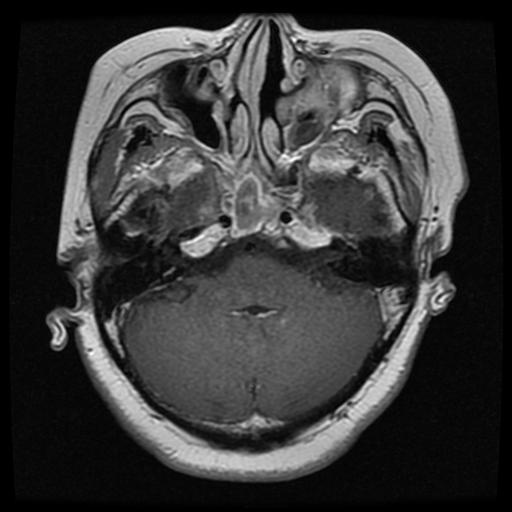
2) EEG brain dataset (Depression dataset).

Load dictionary matrix (D) and apply OMP (optimization algorithm) on each dataset mentioned above and construct sparse representation coefficients of each samples (a1, a2, a3,upto a1024).

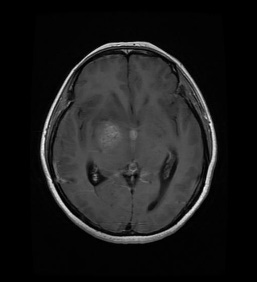
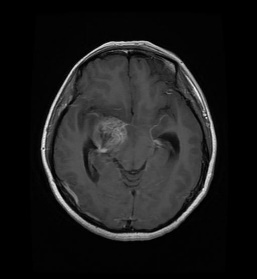
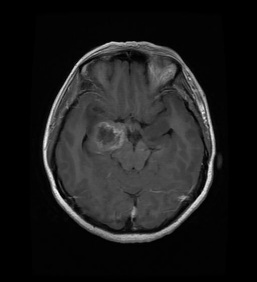
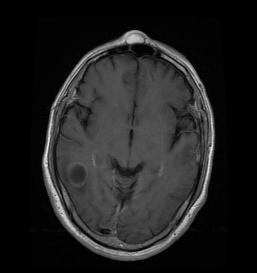
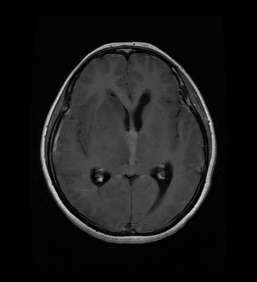
You need to save with zero values and non-zero values or take maximum value. Please see the following function.



Meningioma(708)



pituitary(930)



Glioma(1426)

%% compute sparse cofficents using overcmplete dictionary

D=randn(64,1024); % compute over-complete dictionary with size 200\*1024.

% load Dic\_Dl\_1024\_8\_woSbtMean.mat

% D=Dl;

patch\_size = sqrt(size(D, 1));

I=double(imread('TL.jpg')); % input image

I=I(1:100,1:100);

[m n]=size(I);

Mean\_I1=mean(mean(I))

for ii=1:m-patch\_size+1%:m-patch\_size+1,

count=1;

if (ii>1 && ii<m-patch\_size+1)

for jj = 1:n-patch\_size+1%:n-patch\_size+1,

if (jj>1 && jj<m-patch\_size+1)

patch=I(ii:ii+patch\_size-1, jj: jj+patch\_size-1);

Mean\_patch = mean(patch(:));

patch1=single(patch-Mean\_patch);

sparse\_coeff = SolveOMP(D2, patch1(:), size(D2,2),10);

sparse\_coeff1(jj,count)=max(sparse\_coeff); % store spares coefficients of first image. Similarly, you can store all spare coefficients based on different number of images

end

end

count=count+1;

end

end