```
%% MyMainScript

clear all;

clc;

tic;

%% ORL database

recognize('../../att_faces/',[1, 2, 3, 5, 10, 15, 20, 30, 50, 75, 100, 150, 170],32,6,4);

%% Yale Database

recogyale('../../CroppedYale/',[1, 2, 3, 5, 10, 15, 20, 30, 50, 60, 65, 75, 100, 200, 300, 500,

1000],38,40,24);

toc;

function s=recognize(path,k,nimages,ntrain,ntest)

Dir=dir(path);

start=4;
```

```
lms={Dir(start:end,1).name};

lmpaths=strcat(path,'/',lms(1:end));

for i=1:nimages

    iDir=dir(char(lmpaths(i)));

    for j=1:ntrain

        im= imread(char(strcat(lmpaths(i),'/',iDir(2+j).name)));

        D((i-1)*ntrain+j,:) = im(:);

    end

    for j=3+ntrain:size(iDir,1)

        im= imread(char(strcat(lmpaths(i),'/',iDir(j).name)));

        T((i-1)*ntest+j-2-ntrain,:) = im(:);

    end

end
```

```
D=double(D);

T=double(T);

T = T-repmat(mean(D),size(T,1),1);

D = D-repmat(mean(D),size(D,1),1);

% using eig

% L = D*D';

% [V,ev] = eig(L);

% V=fliplr(V);

[V,s,u] = svd(D);

V = D'*V;

for i=1:size(V,2)

    V(:,i) = V(:,i)./norm(V(:,i));

end
```

```
cD=zeros(size(D,1),size(V,2));

for i=1:size(D,1)

   cD(i,:)=V'*D(i,:)';

end

cT=zeros(size(T,1),size(V,2));

for i=1:size(T,1)

   cT(i,:) = V'*T(i,:)';

end

for kk=1:size(k,2)

    count=0;

    for i=1:ntest*nimages

        for j=1:ntrain*nimages

            nrm(j) = norm(cT(i,1:k(kk))-cD(j,1:k(kk)));
```

```
        end

        [val,ind] = min(nrm);

        if(fix((ind-1)/ntrain) == fix((i-1)/ntest))

            count=count+1;

        end

    end

    result(kk)=(count/(ntest*nimages)) *100;

end

figure;

plot(k,result);

title('Accuracy');

xlabel('no. of eigen faces used');

ylabel('Recognition rate') ;
```

```
function s=recognize(path,k,nimages,ntrain,ntest)

tic;

Dir=dir(path);

start=3;

lms={Dir(start:end,1).name};

lmpaths=strcat(path,'/',lms(1:end));

for i=1:nimages

    iDir=dir(char(lmpaths(i)));

    for j=1:ntrain

        im= imread(char(strcat(lmpaths(i),'/',iDir(2+j).name)));

        D((i-1)*ntrain+j,:) = im(:);

    end

    for j=3+ntrain:size(iDir,1)
```

```
        im= imread(char(strcat(Impaths(i),'/',iDir(j).name)));

        T((i-1)*ntest+j-2-ntrain,:) = im(:);

    end

end

D=double(D);

T=double(T);

T = T-repmat(mean(D),size(T,1),1);

D = D-repmat(mean(D),size(D,1),1);

% L = D*D';

% [V,ev] = eig(L);

% V=fliplr(V);

[V,~,~] = svd(D,'econ');

V = D'*V;
```

```
toc;

for i=1:size(V,2)

    V(:,i) = V(:,i)./norm(V(:,i));

end

cD=zeros(size(D,1),size(V,2));

for i=1:size(D,1)

   cD(i,:)=V'*D(i,:)';

end

cT=zeros(size(T,1),size(V,2));

for i=1:size(T,1)

   cT(i,:) = V'*T(i,:)';

end

for kk=1:size(k,2)
```

```
count1=0;count2=0;

for i=1:ntest*nimages

    for j=1:ntrain*nimages

        nrm1(j) = norm(cT(i,1:k(kk))-cD(j,1:k(kk)));

        nrm2(j) = norm(cT(i,4:3+k(kk))-cD(j,4:3+k(kk)));

    end

    [val,ind1] = min(nrm1);

    [val,ind2] = min(nrm2);

    if(fix((ind1-1)/ntrain) == fix((i-1)/ntest))

        count1=count1+1;

    end

    if(fix((ind2-1)/ntrain) == fix((i-1)/ntest))

        count2=count2+1;
```

```
        end

    end

    result1(kk)=(count1/(ntest*nimages)) *100;

    result2(kk)=(count2/(ntest*nimages)) *100;

end

figure;

plot(k,result1)

title('Accuracy(with all eigen vectors)');

xlabel('no. of eigen faces used');

ylabel('Recognition rate') ;

figure;

plot(k,result2)

title('Accuracy(without first three eigen vectors)');
```

```
xlabel('no. of eigen faces used');

ylabel('Recognition rate') ;

%% MyMainScript

tic;

%% Your code here

reconstruct('../../CroppedYale/',[2, 10, 20, 50, 75, 100, 125, 150, 175],38,40,24);

toc;

function reconstruct(path,k,nimages,ntrain,ntest)

Dir=dir(path);

start=3;

Ims={Dir(start:end,1).name};

Impaths=strcat(path,'/',Ims(1:end));

for i=1:nimages
```

```
    iDir=dir(char(Impaths(i)));

    for j=1:ntrain

        im= imread(char(strcat(Impaths(i),'/',iDir(2+j).name)));

        rows=size(im,1);

        D((i-1)*ntrain+j,:) = im(:);

    end

end

D=double(D);

M=mean(D);

D = D-repmat(M,size(D,1),1);

% L = D*D';

% [V,ev] = eig(L);

% V=fliplr(V);
```

```
[V,~,~] = svd(D,'econ');

V = D'*V;

for i=1:size(V,2)

    V(:,i) = V(:,i)./norm(V(:,i));

end

figure;

for i=1:25

    subplot(5,5,i);

    img=vec2mat(V(:,i),rows);

    imshow(mat2gray(img'));

end

cD(1,:)=V'*D(1,:)';

cD = repmat(cD,size(V,1),1).*V;
```

```
cD=cD';

figure;

for i=1:9

    subplot(2,5,i);

    img=M+sum(cD(1:k(i),:));

    img=vec2mat(img,rows);

    imshow(mat2gray(img'));

%     title(strcat(num2str(k(i)),' eigen faces'));

end

%% MyMainScript

clear all;

clc;

tic;
```

```
%% Stranger Detection

% In the first problem we assumed all test images belong to people in

% training set hence we found out the closest neghbour in eigen subspace.

% Even if the image belonged to stranger it would find the closest match

% Here we are thresholding the closest match and only if the closest match

% is less than the threshold we recognize else we identify the person as a

% stranger

% If the person does not belong to people in training set then his image is

% not likely to match with any traing image (in eigen subspace)

%% Your code here

% repmat([1,2],2,1)

% norm([3,4]')

recognize('../../att_faces/',[5,10,15,20,30,50,75,100,150,170],32,6,4,8,4,3800);
```

```
%% time

toc;

function recognize(path,k,nimages,ntrain,ntest,nout,ntest_out,threshold)

Dir=dir(path);

Ims={Dir(4:end,1).name};

Impaths=strcat(path,'/',Ims(1:end));

for i=1:nimages

    iDir=dir(char(Impaths(i)));

    for j=1:ntrain

        im= imread(char(strcat(Impaths(i),'/',iDir(2+j).name)));

        D((i-1)*ntrain+j,:) = im(:);

    end

    for j=1:ntest
```

```
        im= imread(char(strcat(Impaths(i),'/',iDir(2+j+ntrain).name)));

        T((i-1)*ntest+j,:) = im(:);

    end

end

for i=1:nout

    for j=1:ntest_out

        im= imread(char(strcat(Impaths(i),'/',iDir(2+j).name)));

        T_Out((i-1)*ntest_out+j,:) = im(:);

    end

end

D=double(D);

T=double(T);

T_Out=double(T_Out);
```

```
T = T-repmat(mean(D),size(T,1),1);

D = D-repmat(mean(D),size(D,1),1);

T_Out = T_Out-repmat(mean(T_Out),size(T_Out,1),1);

L = D*D';

[V,ev,U] = svd(D);

V = D'*V;

for i=1:size(V,2)

    V(:,i) = V(:,i)./norm(V(:,i));

end

cD=zeros(size(D,1),size(V,2));

for i=1:size(D,1)

    cD(i,:) = V'*D(i,:)';

end
```

```
cT=zeros(size(T,1),size(V,2));

for i=1:size(T,1)

    cT(i,:) = V'*T(i,:)';

end

cT_Out=zeros(size(T_Out,1),size(V,2));

for i=1:size(T_Out,1)

    cT_Out(i,:) = V'*T_Out(i,:)';

end

for kk=1:size(k,2)

    ecD = double(zeros(size(D,1)));

    for i=1:size(D,1)

       ecD(i) = norm(D(i,:)'-V(:,4:k(kk))*cD(i,4:k(kk))');

    end
```

```
ecT=double(zeros(size(T,1)));

for i=1:size(T)

    ecT(i) = norm(T(i,:)'-V(:,4:k(kk))*cT(i,4:k(kk))');

end

ecT_Out=double(zeros(size(T_Out,1)));

for i=1:size(T_Out,1)

   ecT_Out(i) =norm(T_Out(i,:)'-V(:,4:k(kk))*cT_Out(i,4:k(kk))');

end

count=0;

pos1=0;

err1=0;

err2=0;

pos2=0;
```

```
for i=1:ntest*nimages

    mn = 1000000;

    ind = 0;

        for j=1:ntrain*nimages

            nrm = norm(cT(i,4:k(kk)+3)-cD(j,4:k(kk)+3));

            if(nrm < mn)

                mn = nrm;

                ind = j;

            end

        end

        if(nrm<threshold)

            pos2=pos2+1;

        else
```

```
            err2=err2+1;

        end

        %ntrain

        if(fix((ind-1)/ntrain) == fix((i-1)/ntest))

            count=count+1;

        end

end

for i=1:nout*ntest_out

    mn = 1000000;

    ind = 0;

        for j=1:ntrain*nimages

            nrm = norm(cT_Out(i,4:k(kk)+3)-cD(j,4:k(kk)+3));

            if(nrm < mn)
```

```
                mn = nrm;

                ind = j;

            end

        end

        if(nrm>=threshold)

            pos1=pos1+1;

        else

        %% MyMainScript

clear all;

tic;

%% Your code here

A = rand(150,250);

[U1,S1,V1] = MySVD(A);
```

```
norm(A-U1*(S1*V1'))

toc;

function [U,S,V] = MySVD(A)

[m,n] = size(A);

 k = min(m,n);

U = double(zeros(m,m));

V = double(zeros(n,n));

S = double(zeros(m,n));

if m==n

   [U,D] = eig(A*A');

   V = A'*U;

   for i=1:k

        V(:,i) = V(:,i)./norm(V(:,i));
```

```
        %norm(V(:,i))

    end

end

if m>n

    [Us,Ds] = eig(A*A');

    [d,ind] = sort(diag(Ds),'descend');

    D = Ds(ind,ind);

    U = Us(:,ind);

    V = A'*U(:,1:k);

    for i=1:k

        V(:,i) = V(:,i)./norm(V(:,i));

        %norm(V(:,i))

     end
```

```
end

if m<n

   [Vs,Ds] = eig(A'*A);

   [d,ind] = sort(diag(Ds),'descend');

   D = Ds(ind,ind);

   V = Vs(:,ind);

   U = A*V(:,1:k);

   for i=1:k

      U(:,i) = U(:,i)./norm(U(:,i));

      %norm(V(:,i))

   end

end

S(1:k,1:k) = D(1:k,1:k);
```

```
S = real(sqrt(S));

            err1=err1+1;

        end

        %ntrain

        if(fix((ind-1)/ntrain) == fix((i-1)/ntest))

            count=count+1;

        end

    end

    %[k(kk),100.0*pos1/(pos1+err1),100.0*pos2/(pos2+err2)]

    strcat('k=',num2str(k(kk)),' false positives:',num2str(err1),' false
negatives:',num2str(err2))

end

%% MyMainScript
```

```
clear all;

tic;

%% Your code here

A = rand(150,250);

[U1,S1,V1] = MySVD(A);

norm(A-U1*(S1*V1'))

toc;

function [U,S,V] = MySVD(A)

[m,n] = size(A);

 k = min(m,n);

U = double(zeros(m,m));

V = double(zeros(n,n));

S = double(zeros(m,n));
```

```
if m==n

   [U,D] = eig(A*A');

   V = A'*U;

   for i=1:k

       V(:,i) = V(:,i)./norm(V(:,i));

       %norm(V(:,i))

   end

end

if m>n

   [Us,Ds] = eig(A*A');

   [d,ind] = sort(diag(Ds),'descend');

   D = Ds(ind,ind);

   U = Us(:,ind);
```

```
    V = A'*U(:,1:k);

    for i=1:k

        V(:,i) = V(:,i)./norm(V(:,i));

        %norm(V(:,i))

     end

end

if m<n

    [Vs,Ds] = eig(A'*A);

    [d,ind] = sort(diag(Ds),'descend');

    D = Ds(ind,ind);

    V = Vs(:,ind);

    U = A*V(:,1:k);

    for i=1:k
```

```
        U(:,i) = U(:,i)./norm(U(:,i));

        %norm(V(:,i))

    end

end

S(1:k,1:k) = D(1:k,1:k);

S = real(sqrt(S));
```