PCA 人脸识别

算法描述：

```
function rate=qrPCA(eigenDim,method)
% Train and test
% arg1:eigen dimension=(1:TotalSamples)
% arg2:measure method
% method 1 Manhattan/L1()
% method 2 Euclidian/L2()
% method 3 Mahalanoibis(eigen dimension must less than classes)
[t,ES,EF]=train(eigenDim);
rate=test(t,ES,EF,method);
```

主函数 qrPCA，利用 qr 分解降低计算量的 PCA 人脸识别算法，参数一是设定特征维数，取值应在 1 至训练样本总数，即[1,280]；参数二是选择邻近算法，2 为欧几里得距离，即是作业要求的 2 范数最小匹配，曼哈顿距离以及马氏距离仅作参考（实验中可以发现曼哈顿距离略优于欧氏距离，马氏距离表现不佳，可能因为训练样本太小）。

该函数调用两个子函数，train 和 test。

其中 train 函数是训练样本，参数是选取的特征维数，输出是 t：随机生成的 3 个测试样本的下标，剩余 7 个用于训练，对于 40 个类别这两组数字是使用相同的，所以 t 是一个 1x3 向量；ES：特征空间；EF：特征脸，这里的输出没用按类别取均值，而是每个类别都是 7 个特征脸向量。

Test 函数参数 1-3 是 train 函数的输出，参数 4 是选取的邻近算法。

Train 函数

```
function [t,ES,EF]=train(eigenDim)
```

```
% return test index,eigen space,eigen face
disp('Begin train phase');
tic;
X=zeros();
EigFace=cell(1,40);
classNum=40;
sampleNum=7;
trainIndex=randperm(10,sampleNum);%10 samples per class
testIndex=setdiff((1:10),trainIndex);
t=testIndex;
for i=1:classNum
    for j=trainIndex()

I=reshape(imread(['att_faces/s',num2str(i),'/',num2str(j),'.pgm']),1,
[]);
       if size(X)==[1,1]
          X=I;
       else
          X=[X;I];
       end
    end
end
X=X';
X=double(X);                                %centralise or mean face
H=(X-mean(X,2))/sqrt(classNum*sampleNum-1);     % 1/sqrt(samples - 1)
% S=cell(1,40); %covariance matrix set S=X{i}*X{i}'
disp(['X   ',num2str(size(X))]);
disp(['H   ',num2str(size(H))]);
[Q,R]=qr(H);
[~,D,V]=svd(R');
disp(['Q   ',num2str(size(Q))]);
disp(['R   ',num2str(size(R))]);
disp(['D   ',num2str(size(D))]);
disp(['V   ',num2str(size(V))]);
%     S{i}=Q*V*D'*D*V'*Q';
%     S{i}=H{i}*H{i}';
%     [eigVec eigVal]=eig(S{i});
h=diag(D);
h=h(1:eigenDim);
[vL,~]=size(V);
temp=zeros(vL,eigenDim);
for j=1:eigenDim
    temp(:,j)=V(:,j);
end
```

```
EigSpace=Q*temp;
temp=EigSpace'*X;
for i=1:classNum
    EigFace{i}=temp(:,(i-1)*sampleNum+1:(i-1)*sampleNum+7);
end
ES=EigSpace;
EF=EigFace;
time=toc;
disp(['train time:',num2str(time),' seconds']);
```

PCA 算法流程：

• 将 280 图逐一 reshape 成 1xd 的向量，d 为像素总数（这里的值为 112x92），一个矩阵 X，并转置。

• 计算 X 的行均值 Xm=Σxi，H=X-Xm

• 对 H 进行 QR 分解，H=QR

• 对 R 的转置进行 svd 奇异值分解，R'=UDV'，这里 U 没有储存

• D 得对角元即为 X 的协方差矩阵的特征值，选取若干个特征值对应 的 V 中的列向量组成一个矩阵 Vh，则特征空间就等于 Q*Vh

• 特征空间的转置乘以 X 即为特征脸，这里没有取均值

算法解释：

X 的协方差矩阵 S=H*H'

H=QR

R=UDV'

则 S=QRR'Q'

S=Q(UDV')'UDV'Q'

S=QVD'U'UDV'Q'

由奇异值分解得到的 U 和 V 都是酉矩阵，酉矩阵乘以自身的转置等于单位矩阵，所以：

S=QVD'DV'Q'

由 QR 分解知 Q 也是酉矩阵则：

QVV'Q'=QIQ'=QQ'=I

令 A=QV，B=D'D，则

S=ABA'

SA=ABA'A

SA=ABI=AB

SQV=QVD'D,

D 是对角矩阵，D'D 自然也是对角矩阵，而且非 0 元素的个数和 D 相等，最大非 0 个数为 min(size(D))。取 B 中一个对角元素 b，则上式可变成 Sa=a*b=b*a，其中 b 是一个数，a 是 A 中对应 b 的一个列向量，则说明 B 中非 0 对角元均是 S 的特征值，其对应 A 中列向量则是 A 的特征值对应的特征向量。至此完成了对大矩阵 S 求特征值以及特征向量。

但 S!=cov(X)，这里的 S=cov(X')，两幅人脸的协方差没有用，PCA 使用的协方差是对应位置的像素点之间的协方差。

Test 函数

```
function rate=test(t,ES,EF,method)
% method 1 Manhattan/L1
% method 2 Euclidian/L2
% method 3 Mahalanoibis(eigen dimension must less than sampleNum(7))
```

```
disp('Begin test phase');
rate=0;
classNum=40;
mEF=cell(1,40);
if ismember(method,[1,2,3])
    for i=1:classNum
        mEF{i}=mean(EF{i},2);
    end
    for i=1:classNum
        for k=1:3
     I=imread(['att_faces/s',num2str(i),'/',num2str(t(k)),'.pgm']);
            I=reshape(I,[],1);
            I=double(I);
            y=ES'*I;
            sum=zeros(classNum,1);
            if method == 1
                for l=1:classNum
                    sum(l)=norm(mEF{l}-y,1);%Manhattan L1
                end
            elseif method ==2
                for l=1:classNum
                    sum(l)=norm(mEF{l}-y,2);%Euclidian L2
                end
            elseif method == 3
                for l=1:classNum
%                     sum(l)=mahal(y',EF{l}');
                    temp=cov(EF{l}')^-1;
                    sum(l)=(y-mEF{l})'*temp*(y-mEF{l});
                end
            end
            c=find(sum==min(sum));
            if i==c
                rate=rate+1;
                %disp(['s',num2str(i),'/',num2str(t(k)),'.pgm is
matched.']);
            else
                disp(['s',num2str(i),'/',num2str(t(k)),'.pgm is not
matched. Wrong ans is ',num2str(c),'.']);
            end
        end
    end
    disp(['ratio:',num2str(rate/1.2),'%']);
else
    disp('wrong method number,try help');
```

```
end
```

先把特征脸的按类别算均值得到 mEF，然后在逐一读取 3x40 张测试样本，对测样本 reshape 成一个向量，用特征空间的逆乘以该向量，得到一个长为选取的特征维数的向量 y，y 即是该图像在特征空间的投影，然后按选取的测量方法使用对应的公式计算出 y 和 40 个类别的邻近距离，取最小距离的类别作为 y 的类别，类别如果等于当前文件夹的序号则说明匹配准确。L1 范数即是曼哈顿距离，L2 范数即是欧几里得距离，马氏距离中使用了未取平均的特征脸矩阵的协方差矩阵的逆，在一个类别的训练样本数（这里就是 7）大于特征维数的时候才能正常计算，而小于或等于时计算出来的协方差的逆矩阵误差太大并没有实用价值，matlab 的提示是这个逆矩阵接近或等于奇异矩阵。所以选取方法 3 的时候只能选取不大于 6 个特征维，但实际匹配真确率太低，可能是训练样本太少，导致最多能选的 6 个最大特征根保留了太少主成分。

欧几里得距离的 PCA 数据测试

h	**55**	**60**	**70**	**75**	**80**	**85**	**90**	**95**	**100**	**105**
1	112	108	109	114	111	109	109	112	114	108
2	111	107	109	113	111	110	112	108	107	115
3	106	106	114	114	111	107	111	111	113	112
4	107	110	110	107	111	111	109	116	107	112
5	113	108	112	111	115	111	108	110	114	111
mean	109.8	107.8	110.8	111.8	111.8	109.6	109.8	111.4	111	111.6
ratio	0.915	0.898	0.923	0.931	0.931	0.913	0.915	0.928	0.925	0.93

h 为选取的特征维数，对每一个选取的特征维数进行五次测试，实用平均值作为结果。准确率在 92%左右。测试时间在 5 分钟左右。