

# 基于 K-L 变换的人脸识别应用

实验人员：

2009302 袁 嗽

2009302323 彭群 聂

2009302309 吕 鑫

## 一. 实验要求

1、从网上下载人脸图像，构建人脸训练数据库和测试数据库，采用 K-L 变换进行特征脸提取，并实现人脸识别。通过 K-L 变换在人脸识别中的应用，加深对所学内容的理解和感性认识。

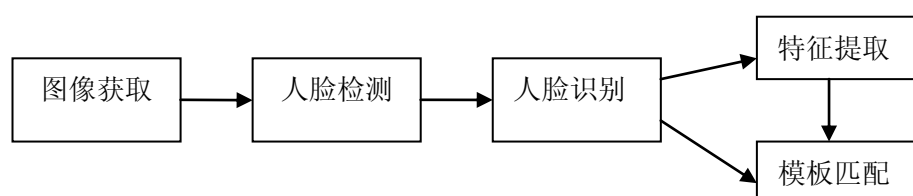
2、或者从网上下载其它数据库，编程实现 K-L 变换。

## 二. 实验目的

1. 通过 K-L 变换在人脸识别中的应用实验加深对所学知识的印象，所学内容的理解和感性认识。
2. 加强理论知识与实践的结合，将理论运用到实际中，同时在实验过程中巩固理论知识

## 三. 实验原理及准备知识

1. 人脸识别指的是给定一个场景中的静态图像或者视频，利用给定的人脸数据库信息，鉴别该场景中一位或多位人的过程。主要分以下几步：



2. 本实验采用的是基于特征脸的方法进行识别，特征脸的方法则是建立在 K-L 变换之上的。高维的图像空间经过 K-L 变换之后得到一组新的正交基，保留其中重要的正交基，这些正交基可以构成低维的线性空间，实现降维的目的。如果假设人脸在这些低维线性空间中的投影具有可分性，就可以将这些投影用作识别的特征向量，即特征脸。
3. 平均脸指的是所有人脸图像的平均向量。
4. K-L 变换的思想：利用 K-L 变换确定相应的人脸基图像，再反过来用这些基图像对人脸图像库中的所有人脸图像进行 K-L 变换，从而得到每幅图

像的参数向量并将每幅图的参数向量保存下来。在识别时，先对一张所输入的脸图像进行必要的规范化，再进行 K-L 变换分析，得到其参数向量。将这个参数向量与库中每幅图的参数向量进行比较，找到最相似的参数向量，也就等于找到最相似的人脸。

## 五. 实验具体步骤

1. 读入训练人脸库。读入时，将每个人脸转化为一维的向量，假设共有  $N$  个人脸图像，每幅图像的大小为  $m \times n$ （表示图像宽和高），则训练图像集可以表示为一个  $(m \times n) \times N$  的矩阵。

2. 其中第  $i$  幅人脸可以表示为：

$$x^i = [x_1^i, x_2^i, \dots, x_{m \times n}^i] (i = 1, 2, \dots, N) \quad (4-1);$$

3. 计算平均脸，以及每张人脸到平均脸的距离。
4. 平均脸的计算方法：

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x^i \quad (4-2)$$

则每张人脸和平均脸之间的距离向量可以表示为：

$$d_i = x^i - \bar{x} \quad (4-3)$$

5. 计算特征脸。

$$\text{设矩阵} \quad A = [d_1, d_2, \dots, d_N] \quad (4-4)$$

$$\text{协方差矩阵为:} \quad \frac{1}{N} A A^T = \frac{1}{N} \sum_{i=1}^N d_i d_i^T \quad (4-5)$$

$$\text{其中 } A A^T \text{ 的转置为:} \quad A^T A \quad (4-6)$$

计算特征脸时需要求出协方差矩阵的前  $d$  ( $d \ll N$ ) 个特征值。由于式 (4-5) 是一个  $(m \times n) \times (m \times n)$  的高维向量，其计算量很大，而式 (4-6) 是一个  $N \times N$

的低维向量，因此可以先求出式 (4-6) 的值：

$$v_j = A u_j \lambda_j^{-1/2} \quad (j = 1, 2, \dots, d) \quad (4-7)$$

其中  $u_j$  是式 (4-6) 的特征向量， $\lambda_j$  是式 (4-5) 的特征值。

则，式 (4-7) 即为所需要的特征脸。

式 (4-7) 可以组成一个特征脸空间：

$$W = \{v_1, v_2, \dots, v_d\} \quad (4-8)$$

## 6. 投影

将每一幅人脸与平均脸的差值矢量投影到“特征脸”空间，可得第*i*个特征脸向量表示为：

$$P_i = W^T d_i \quad (i=1, 2, \dots, N) \quad (4-9)$$

## 7. 识别（利用欧氏距离分类）。

距离阈值为：

$$\theta = \frac{1}{2} \max_{j,k} \{\|P_j - P_k\|\} \quad (j, k = 1, \dots, N) \quad (4-10)$$

把待识别的人脸 R 投影到特征脸空间，得到向量：

$$P = W^T (R - \bar{x}) \quad (4-11)$$

P 与每个人脸集得距离定义为：

$$\varepsilon_k^2 = \|P - P_k\|^2 \quad (k = 1, 2, \dots, N) \quad (4-12)$$

为了区分人脸与非人脸，还需要计算原始图像 R 和由特征脸空间重建的图像  $R_f$  之间的距离  $\varepsilon$ ：

$$\varepsilon^2 = \|R - R_f\|^2 \quad (4-13)$$

$$R_f = WP + \bar{x} \quad (4-14)$$

则有：判决规则如下：

- (1) 若  $\varepsilon \geq \theta$ ，则输入图像不是人脸图像；
- (2) 若  $\varepsilon < \theta, \forall k, \varepsilon_k \geq \theta$ ，则输入图像包含未知人脸；
- (3) 若  $\varepsilon < \theta, \forall k, \varepsilon_k < \theta, \varepsilon_i = \min\{\varepsilon_k\}$ ，则输入图像为库中第 i 个人的脸。

## 六.实验流程图

## 七. 实验结果及分析

本实验中，选取 15 个人的 5 组表情(也就是 75 张人脸)作为训练样本集，也就是作为模板，选取此 15 个人的其余 6 组表情作为测试样本集，即待识别样本集。

### 1. 对训练样本集进行测试

选取库中的 2 张人脸（即第 2 人的第 1 种表情）：



PCA  
人脸识别系统

识别结果

2



训练样本集

s1 (1)

s1 (2)

s1 (3)

s1 (4)

训练集测试

测试样本集

s6 (1)


s6 (2)

s6 (3)

s6 (4)

测试集测试


第 2 人的第 5 种表情：



PCA  
人脸识别系统

识别结果

2



训练样本集

s5 (2)

s5 (3)

s5 (4)

s5 (5)

训练集测试

测试样本集

s6 (1)

s6 (2)

s6 (3)

s6 (4)

测试集测试

第 8 人的第 1 种表情



第 8 人的第 5 种表情：



结论分析：

可以看出训练样本集中人脸的识别正确率是比较理想的，不管是不同的表情或是不同的人，它都能准确的识别出来。这就说明对于训练集来讲该识别算法可信度很高。

## 2. 对测试集的识别结果:

选取第 3 人的第 6 种表情:



PCA  
人脸识别系统



识别结果 3

训练样本集

s5 (5)  
s5 (6)  
s5 (7)  
s5 (8)

训练集测试

测试样本集

s6 (1)  
s6 (2)  
s6 (3)  
s6 (4)

测试集测试

选取第 15 人的第 6 种表情



PCA  
人脸识别系统



识别结果 15

训练样本集

s5 (5)  
s5 (6)  
s5 (7)  
s5 (8)

训练集测试

测试样本集

s6 (13)  
s6 (14)  
s6 (15)  
s7 (1)

测试集测试

第 15 人的第 8 种表情

PCA  
人脸识别系统



识别结果 15

训练样本集

s5 (5)  
s5 (6)  
s5 (7)  
s5 (8)

训练集测试

测试样本集

s8 (12)  
s8 (13)  
s8 (14)  
s8 (15)

测试集测试

第 15 人的第 10 种表情

PCA  
人脸识别系统



识别结果 8

训练样本集

s5 (5)  
s5 (6)  
s5 (7)  
s5 (8)

训练集测试

测试样本集

s10 (13)  
s10 (14)  
s10 (15)  
s11 (1)

测试集测试

## 结果分析

在测试集的识别中，对于第 15 人，前两种表情都能够正确的识别出来，而第 10 种表情却出现了错误，这说明对于实际中变化多样的人脸表情来讲，该识别算法并不能达到百分百的正确率，存在一定的错误，主要是因为实际获取的人脸图像中外界影响因素太多，如光线问题，角度问题，等等都



会导致识别出现错误。但应该肯定的是，总体上的正确率还是令人满意的。

## 八. 实验感想

本次试验难度要比前两次大，工作量也更大。不过有了前两次的实验的基础，我们还是能够按部就班的将实验进行下去，并最终获得满足实验基本要求的实验结果。

本次试验的收获有：首先，对于基于 PCA 的人脸识别方法有了一定掌握，同时，对于 K-L 变换的思想有了更深入的了解；第二，由于本次试验与实际结合比较紧密，也锻炼了我们如何用所学知识解决实际问题的能力。

## 附件

```
function listBox1_Callback(hObject, eventdata, handles)
str=get(handles.listBox1,'string');
v=get(handles.listBox1,'value');
a=[str{v} '.bmp'];
axes(handles.axes1)
imshow(a);
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
N=75;
q=5;
x=zeros(10000,15);
for i=1:N
    a=[num2str(i) '.bmp'];
    h=imread(a);
    [m n]=size(h);
    x(:,i)=reshape(h,m*n,1);
end
```

```
pingjun=mean(x,2);
d= repmat(pingjun,1,N);
A=x-d;
y=A'*A;
[v u]=eig(y);
tezhengzhi=sum(u);
[tezhengzhi,xulie]=sort(tezhengzhi,2,'descend');
```

```

for i=1:q
tezhenglian(:,i)=A*v(:,xulie(i))*(tezhengzhi(i)^(-0.5));%  $\frac{1}{\sqrt{A}}$ 
end

```

```

for i=1:N
P(:,i)=tezhenglian'*A(:,i);
end
str=get(handles.listbox1,'string');
v=get(handles.listbox1,'value');
a=[str{v} '.bmp'];
z=zeros(10000,1);
h=imread(a);
[m n]=size(h);
z(:,1)=reshape(h,m*n,1);
shibie=tezhenglian*(z-pingjun);%  $\frac{1}{\sqrt{A}}$ 

```

```

chonggou=tezhenglian*shibie+pingjun;%  $\frac{1}{\sqrt{A}}$ 
fangcha=((z-chonggou)*(z-chonggou))^0.5;
yuzhi=0;
for i=1:N
    for j=i:N
        zanshiyuzhi=((P(:,i)-P(:,j))*(P(:,i)-P(:,j)))^0.5;
        if zanshiyuzhi>yuzhi
            yuzhi=zanshiyuzhi;
        end
    end
end
yuzhi=yuzhi/2;

```

```

juli=9e+009;
for i=1:N
    bijiao=((shibie-P(:,i))*(shibie-P(:,i)))^0.5;
    if bijiao<juli;
        juli=bijiao;
        k=i;
    end
end
end

```

```

if fangcha>=yuzhi flag=1;
elseif fangcha<yuzhi&& juli>=yuzhi flag==2;
elseif fangcha<yuzhi&& juli<yuzhi flag=3;
end
if flag==1
    a=[ '0.png'];
    axes(handles.axes3)
    imshow(a);
    set(handles.edit1,'string','Î±»Ê ¶±ð£¬Ç äÖØÐÂ Ê¼ ¬');
elseif flag==2
    a=[ '0.png'];
    axes(handles.axes3)
    imshow(a);
    set(handles.edit1,'string','Ê äÈ ë¼ Ì Ñ ü °¬ Î Õ ÈË Á ³);
elseif flag==3
    if k>15
        ren=rem(k-1,15)+1;
    else ren=k;
    end
    a=[num2str(k) '.bmp'];
    axes(handles.axes3)
    imshow(a);
    set(handles.edit1,'string',ren)
end

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in listbox3.
function listbox3_Callback(hObject, eventdata, handles)
str=get(handles.listbox3,'string');
v=get(handles.listbox3,'value');
a=[str{v} '.bmp'];
axes(handles.axes1)
imshow(a);

```

```

% --- Executes during object creation, after setting all properties.
function listbox3_CreateFcn(hObject, eventdata, handles)
str=get(handles.listbox3,'string');

```

```

v=get(handles.listbox3,'value');
a=[str{v} '.bmp'];
axes(handles.axes1)
imshow(a);

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
N=75;
q=5;
x=zeros(10000,15);
for i=1:N
    a=[num2str(i) '.bmp'];
    h=imread(a);
    [m n]=size(h);
    x(:,i)=reshape(h,m*n,1);
end

pingjun=mean(x,2);
d= repmat(pingjun,1,N);
A=x-d;
y=A'*A;
[v u]=eig(y);
tezhengzhi=sum(u);
[tezhengzhi,xulie]=sort(tezhengzhi,2,'descend');

for i=1:q
    tezhenglilian(:,i)=A*v(:,xulie(i))*(tezhengzhi(i)^(-0.5));%  $\frac{1}{\sqrt{\lambda_i}}$ 
end
for i=1:N
    P(:,i)=tezhenglilian'*A(:,i);
end
str=get(handles.listbox3,'string');
v=get(handles.listbox3,'value');
a=[str{v} '.bmp'];
z=zeros(10000,1);
h=imread(a);

```

```

[m n]=size(h);
z(:,1)=reshape(h,m*n,1);
shibie=tezhenglian*(z-pingjun);% 计算平均值

chonggou=tezhenglian*shibie+pingjun;% 计算重合度
fangcha=((z-chonggou)*(z-chonggou))^0.5;
yuzhi=0;
for i=1:N
    for j=i:N
        zanshiyuzhi=((P(:,i)-P(:,j))*(P(:,i)-P(:,j)))^0.5;
        if zanshiyuzhi>yuzhi
            yuzhi=zanshiyuzhi;
        end
    end
end
yuzhi=yuzhi/2;

juli=9e+009;
for i=1:N
    bijiao=((shibie-P(:,i))*(shibie-P(:,i)))^0.5;
    if bijiao<juli;
        juli=bijiao;
        k=i;
    end
end

if fangcha>=yuzhi flag=1;
elseif fangcha<yuzhi&&juli>=yuzhi flag==2;
elseif fangcha<yuzhi&&juli<yuzhi flag=3;
end
if flag==1
    a=[ '0.png'];
    axes(handles.axes3)
    imshow(a);
    set(handles.edit1,'string','计算平均值');
elseif flag==2
    a=[ '0.png'];
    axes(handles.axes3)
    imshow(a);
    set(handles.edit1,'string','计算重合度');
elseif flag==3
    if k>15

```

```
        ren=rem(k-1,15)+1;
    else ren=k;
end
a=[num2str(k) '.bmp'];
axes(handles.axes3)
imshow(a);
set(handles.edit1,'string',ren)
end
```