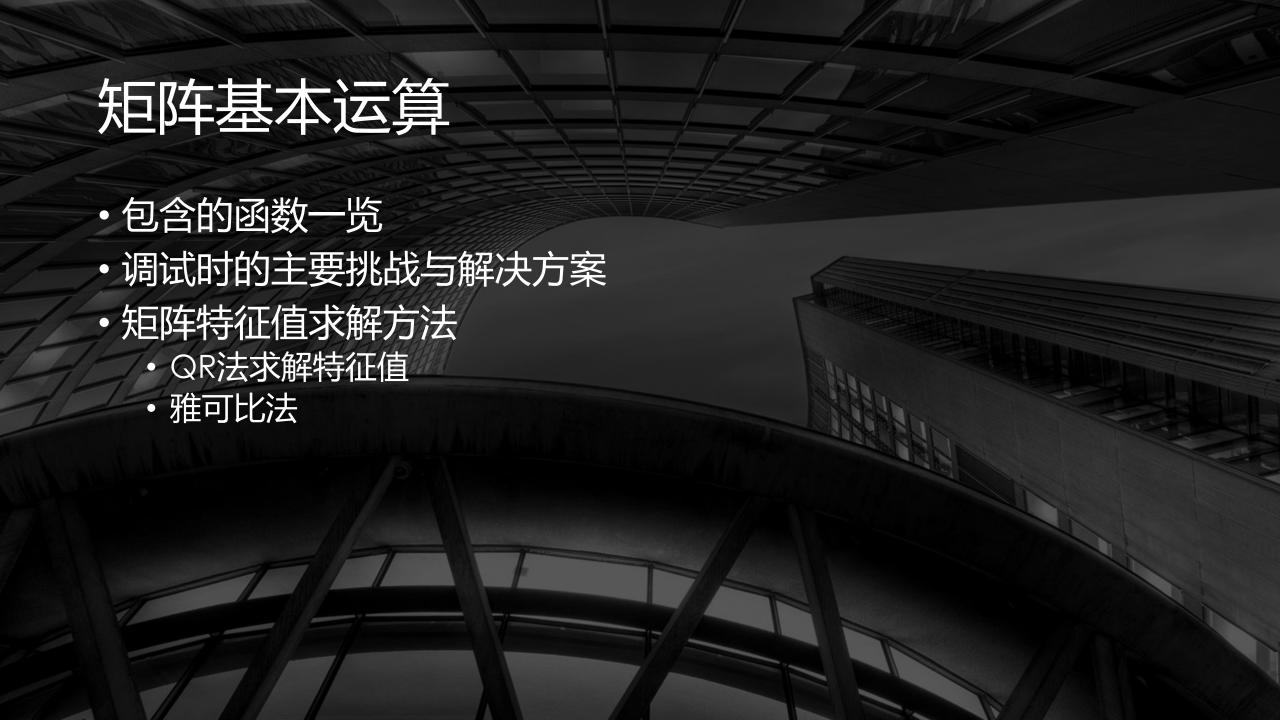
TEAM REPORT 计算机图形学小组展示

人脸识别之特征脸算法

小组成员

李鹏飞、陈欢、胡涵、林葳洁、王凯笛





包含的函数一览

```
#include <stdio.h>
    #include <stdlib.h>
   #include <math.h>
   #define Inf1 1e-5
 6 typedef double ElementType;
   struct Matrix { /* 队列中的结点 */
       ElementType **Data;
       int row;
       int column;
   int CreateMatrix(ElementType *a,int row,int column,struct Matrix *A);//根据二维数组新建矩阵
   void DisplayMatrix(struct Matrix A);// 显示矩阵
16 int CopyMatrix(struct Matrix A, struct Matrix *B);// 复制矩阵
   int AddMatrix(struct Matrix A, struct Matrix B, struct Matrix *C); // 矩阵相加
18 int MinusMatrix(struct Matrix A, struct Matrix B, struct Matrix *C); //矩阵相減
19 int MultiplyMatrix(struct Matrix A, struct Matrix B, struct Matrix *C);// 矩阵相乘
20 int TransposeMatrix(struct Matrix A, struct Matrix *B);// 矩阵转置
21 int MergeMatrix(struct Matrix A, struct Matrix *B); //矩阵合并
22 int DeleteMatrix(struct Matrix *A); // 矩阵删除
23 int CreateIdentityMatrix(struct Matrix *A,int size);// 新建单位阵
24 int JudgeSymmetry(struct Matrix A);// 判断对称性
25 int FindSMatrixElement(struct Matrix A, int *row, int *column);//寻找 最大元素,只用于对称矩阵情况
26 int StandradMatrixR(struct Matrix A, struct Matrix *B);// 对于行向量,进行标准化
27 int StandradMatrixC(struct Matrix A, struct Matrix *B);// 对于列向量,进行标准化
28 int SwapBuffer(struct Matrix A, struct Matrix *B); // 手动交换空间
29 int EigMatrix(struct Matrix A, struct Matrix *B, struct Matrix *C,int Nmax,double prec);//求取特征值函数
    int FindOrder(struct Matrix A, double *order);// 按照大小顺序排列特征值
   int main()
```

调试时的主要挑战与解决方案

- •问题一:使用数组生成矩阵struct,指针存在问题
 - 将二位数组指针强制类型转换,找到每行的头指针,然后再初始化二级 指针
- •问题二:函数内产生的中间矩阵,由于无数组数据,无法通过自己编写的CreateMatrix来创建,同时中间结果无法自动Swap
 - 编写SwapBuffer函数,手动交换缓冲区数据
- •问题三:由于人脸图像数据比较大,因此可能会造成内存不够而导致malloc无法成功
 - 编写DeleteMatrix函数来手动删除中间数据,释放空间



• QR法

QR方法在特征值计算问题的发展上具有里程碑意义。在 1955年的时候人们还觉得特征值的计算是十分困扰的问题,到 1965年它的计算——基于QR方法的程序已经完全成熟。直到今天 QR方法仍然是特征值计算的有效方法之一。

QR算法的基本思想

记 $A=A_1$ 且有 $A_1=Q_1R_1$ (注: Q为正交矩阵,R为上三角矩阵) 将等号右边两个矩阵因子的次序交换,得 $A_2=R_1Q_1$

 $A_2 = Q_1^{-1}A_1Q_1$ 即 $A_2 \sim A_1$ 可证明 $A_{k+1} = Q_k^{-1}A_kQ_k = Q_k^{-1}.....Q_1^{-1}A_1Q_1.....Q_k$

综上: 矩阵序列{A_k}有相同的特征值

QR算法的基本思想

如果A是一个满秩的上Hessenberg矩阵,可以证明,经过一个QR迭代步得到的 $A_2 = Q^{-1}_1 A_1 Q_1$ 仍然是上Hessenberg矩阵。

因为上Hessenberg矩阵次对角线以下的元素全为0,当k→∞

时,由迭代格式产生的矩阵Ak的次对角元趋向于零。

算法优点:

可计算非对称的矩阵

算法缺点:

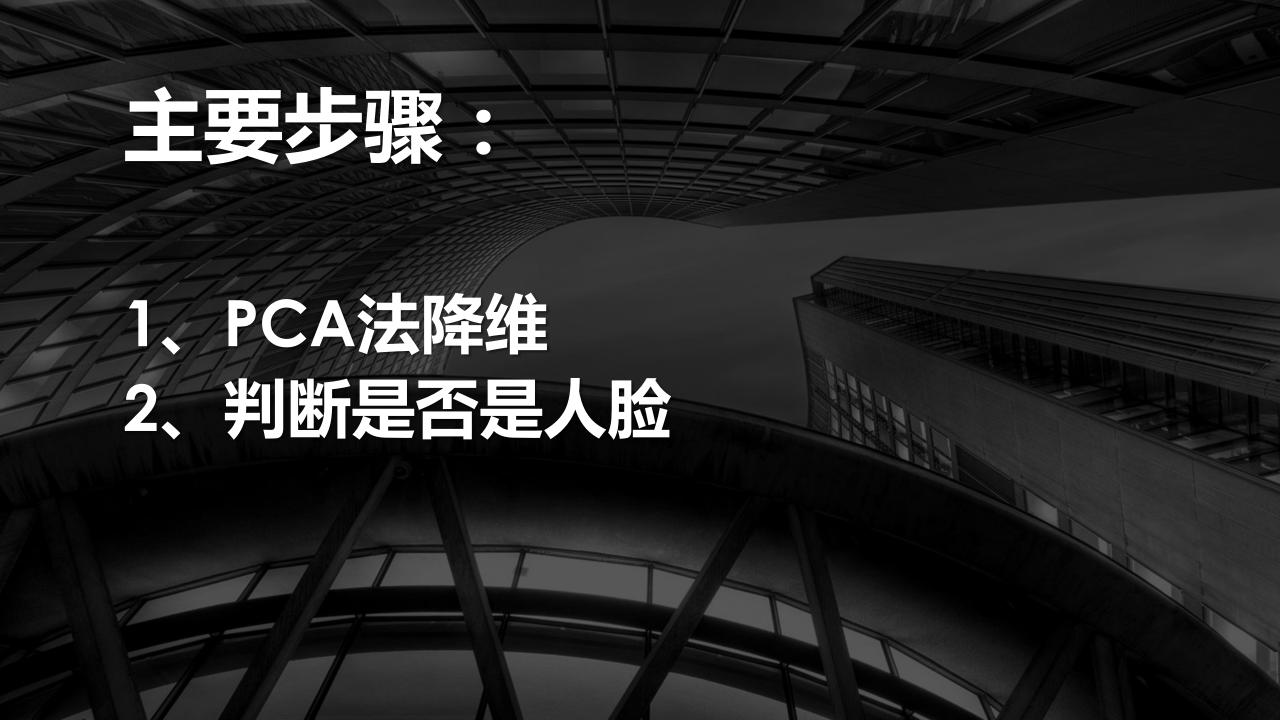
运算复杂度高

Jacobi法求特征向量

与QR法类似,也是做迭代,因此需要做精度控制,满足精度的要求时,退出循环

- •中心思想:
 - 在保证特征值不变的条件下, 使主对角线外元素趋于零
- •实现方法:
 - 寻找最大非对角元素, 乘一个正交的矩阵, 使其变为零
- 迭代控制:
 - 寻找最大的非对角元元素,使其小于设定阈值,使得算法收敛
- 算法不足之处:
 - 只能计算对称矩阵





协方差矩阵的计算

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1(p-1)} & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2(p-1)} & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{(n-1)1} & x_{(n-1)2} & \cdots & x_{(n-1)(p-1)} & x_{(n-1)p} \\ x_{n1} & x_{n2} & \cdots & x_{n(p-1)} & x_{np} \end{pmatrix}$$
标准化

$$A = \begin{pmatrix} x_{11} - \overline{x_1} & x_{12} - \overline{x_2} & \cdots & x_{1(p-1)} - \overline{x_{p-1}} & x_{1p} - \overline{x_p} \\ x_{21} - \overline{x_1} & x_{22} - \overline{x_2} & \cdots & x_{2(p-1)} - \overline{x_{p-1}} & x_{2p} - \overline{x_p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{(n-1)1} - \overline{x_1} & x_{(n-1)2} - \overline{x_2} & \cdots & x_{(n-1)(p-1)} - \overline{x_{p-1}} & x_{(n-1)p} - \overline{x_p} \\ x_{n1} - \overline{x_1} & x_{n2} - \overline{x_2} & \cdots & x_{n(p-1)} - \overline{x_{p-1}} & x_{np} - \overline{x_p} \end{pmatrix}$$

平均脸

特征脸的计算

 A^TA 的特征向量 ------> 求解 AA^T 的特征向量 $v_1, v_2, \cdots v_n$ 与特征值 $u_1, u_2, \cdots u_n$ $-----\rightarrow A^Tv_1, A^Tv_2 \cdots A^Tv_n$ 为 A^TA 所对应的**特征向量**, 特征值仍然为 $u_1, u_2, \cdots u_n$ $A^Tv_1, A^Tv_2 \cdots A^Tv_n$ 标准化后依次序即为所求的主成分,即为特征脸,记为 $u_1, u_2 \cdots u_n$











前10张特征脸











图像在主成分上的投影

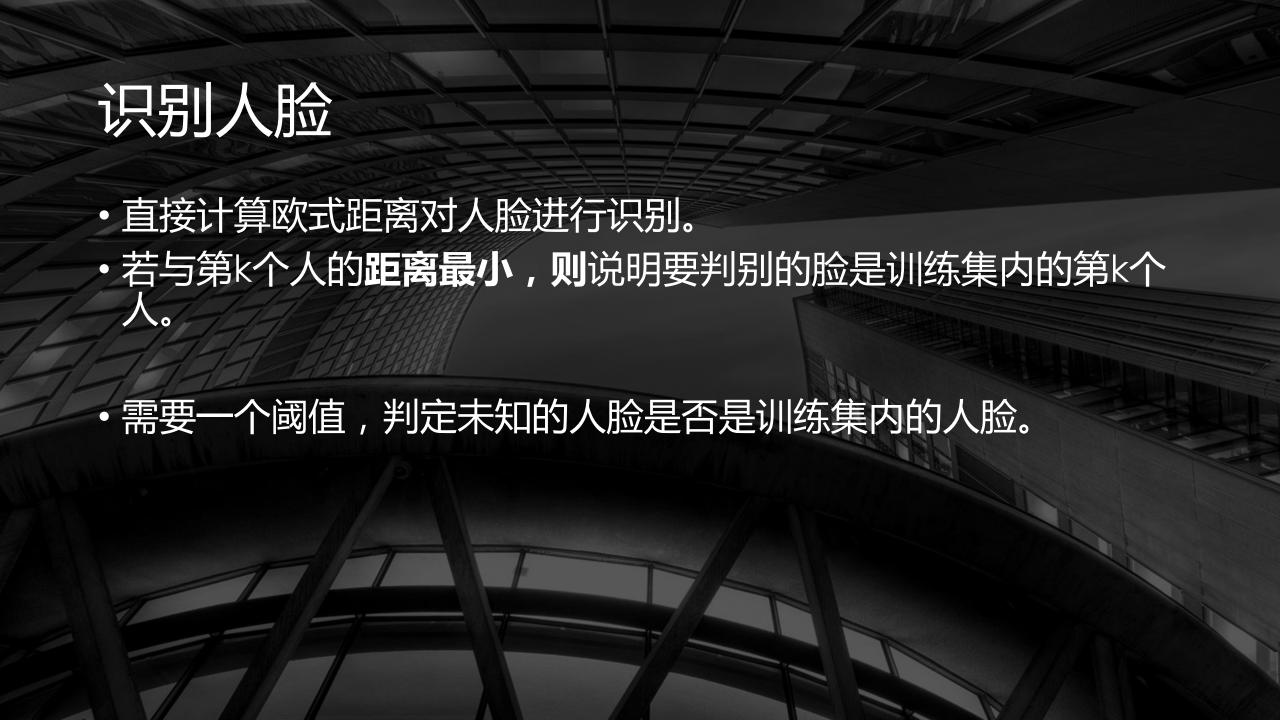
记Γ为输入的图像向量, ψ 为所有训练图像的平均值,则有 $\omega_k = u_k^T (\Gamma - \psi)$ 表示在第k个主成分上的投影的坐标,恢复图像的过程中,只需要选取前M' = 20个主成分进行恢复,即可以得到相应的解压缩以后的图像,注意,在解压缩过程中应该将平均向量最后加回,此时图像的压缩幅度为515.2倍。



识别人脸

- 考虑一张未知的人脸(N维)
- 可以得到它的M维权重(M是特征脸的数目)
- 计算方法: 先与平均脸求差值, 再转化到M维空间(分别与各个特征向量相乘即可)

$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

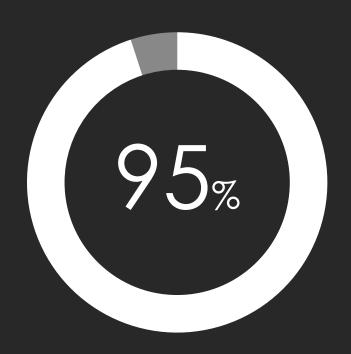


Result.txt >

- (Min Distance=1339.414517, No.5; Max Distance=6200.454682, No.1)
- It should be No.5, and it seems to be No.5. (Min Distance=866.114481, No.5; Max Distance=6251.606045, No.1)
- It should be No.6, and it seems to be No.6.
- (Min Distance=669.224653, No.6; Max Distance=5166.725333, No.5)
- It should be No.6, and it seems to be No.6. (Min Distance=1621.974936, No.6; Max Distance=4994.115345, No.5)
- It should be No.7, and it seems to be No.7. (Min Distance=1139.934853, No.7; Max Distance=5792.531164, No.5)
- It should be No.7, and it seems to be No.7. (Min Distance=1555.657521, No.7; Max Distance=5413.554706, No.5)
- It should be No.8, and it seems to be No.8. (Min Distance=931.627973, No.8; Max Distance=4715.969046, No.7)
- It should be No.8, and it seems to be No.8. (Min Distance=1414.231639, No.8; Max Distance=5020.762879, No.7)
- It should be No.9, and it seems to be No.9. (Min Distance=2155.783896, No.9; Max Distance=5020.521883, No.5)
- It should be No.9, and it seems to be No.7.
 (Min Distance=2899.046451, No./; Max Distance=5465.005484, No.5)

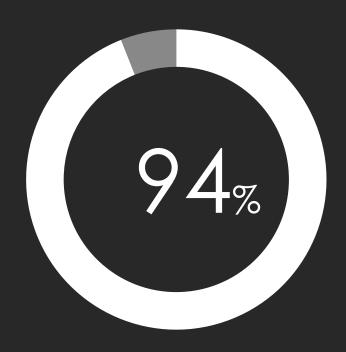
测试过程: 取人脸库库内的非训练集 图片70张进行测试。 将测试库分为A(20张), B(50张)。

测试结果





・ 正确率:19/20



· 输入库B内的图片,用人脸识别算法,检验是否 为库A内的人脸。

• 正确率:47/50

Matlab代码验证精度

>使用函数 pca

```
close all;
        clear:
        fileRoot = './FIGURE/':
        fileIn = fullfile(fileRoot, '*.pgm');
        fileOut = './OUT/':
        files = dir(fileIn);
        fileSize = size(files, 1):
        [im m, im n] = size(imread(fullfile(fileRoot, files(1).name)));
        imgs = {}:
10 -
        disp('read images');
11 -
        disp('...');
12 -
       ☐ for i = 1:fileSize
13 -
            imgs {i} = reshape(mat2gray(imread(fullfile(fileRoot, files(i).name))), 1, []);
14 -
15 -
       end
        disp('read images done!');
16 -
19 -
        imgs = double(cell2mat(imgs')):
        meanImg = mean(imgs);
20
        imgs= imgs - repmat(meanImg, fileSize, 1);
        disp('pca processing');
        disp('...');
23 -
        [coef, score, latent, t2] = pca(imgs);%主成分分析
24 -
        outImgs = coef(:, 1:20);
25 -
26 -
        disp('pca done!');
      for j = 1:size(outImgs, 1)
            img = mat2gray(reshape(outImgs(j, :), im_m, im_n));
28 -
29
            figure, imshow(img);
30
            imwrite(img, strcat(fileOut, 'pca_', int2str(j), '.pgm'));
31 -
        disp(strcat('write image to:', fileOut));
32 -
        disp('end');
33 -
```







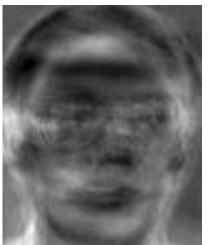


结果展示

> 特征脸(部分)











. Is it anyone in the library: Yes

• It seems to be: 7

• -----

• 文件名: 7l.pgm

路径: C:\Users\Administrator\Desktop\hh_img_complete\pngs\71.pgm

• 修改时间: 1994-04-18 14:07:14

• 体积: 10.08 KB

5l.pgm



• Is it anyone in the library: Yes

• It seems to be: 5

• -----

• 文件名: 5l.pgm

 路径: C:\Users\Administrator\Desktop \hh_img_complete\pngs\5l.pgm

• 修改时间: 1994-04-18 14:07:10

• 体积: 10.08 KB

. Is it anyone in the library: Yes

• It seems to be: 5

• -----

• 文件名: 5J.pgm

 路径: C:\Users\Administrator\Desktop \hh_img_complete\pngs\5J.pgm

• 修改时间: 1994-04-18 14:07:12

• 体积: 10.08 KB

4l.pgm



. Is it anyone in the library: Yes

• It seems to be: 4

• -----

• 文件名: 4l.pgm

 路径: C:\Users\Administrator\Desktop \hh_img_complete\pngs\41.pgm

• 修改时间: 1994-04-18 14:07:08

• 体积: 10.08 KB

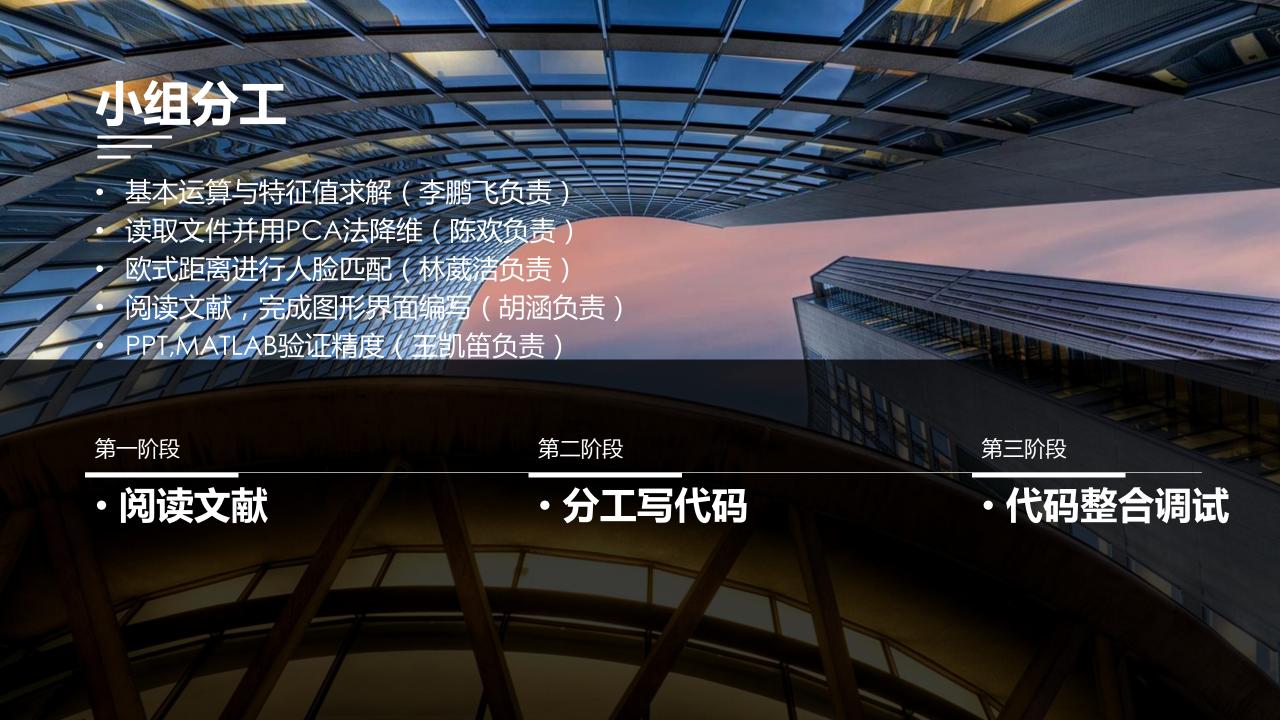
3J.pgm

结果显示界面

展示了待检测图像及其相关信息,以及经过人脸识别程序判断的结果。

运行程序

打开网页



Thank you 感谢观看