**Matlab基于PCA算法的简单图像人脸识别**

（Untitled.m主要调用及运行函数，调用PCA.m、imgdata.m、find.m实现人脸图片读取、识别、输出功能，此方法识别准确率低。）

**Untitled.m：**

img=imgdata(); %图片矩阵数据

Cell\_ten=PCA(img,2);% PCA

face1=find(Cell\_ten,imread('test.jpg'));% find.m识别（参考大佬方案）

subplot(1,2,1)

imshow('test.jpg')

title('选取图像')

subplot(1,2,2)

imshow(strcat(num2str(face1),'.jpg'))

title('匹配图像')

**imgdata.m：**

输入：人脸图片

输出：图片矩阵

function ImgData = imgdata()

namud = 0.5; %原始图片缩小倍数

%导入图片

pic1= rgb2gray(imread('1.jpg')); pic1 = imresize(pic1,namud);

pic2= rgb2gray(imread('2.jpg')); pic2 = imresize(pic2,namud);

pic3= rgb2gray(imread('3.jpg')); pic3 = imresize(pic3,namud);

pic4= rgb2gray(imread('4.jpg')); pic4 = imresize(pic4,namud);

pic5= rgb2gray(imread('5.jpg')); pic5 = imresize(pic5,namud);

pic6= rgb2gray(imread('6.jpg')); pic6 = imresize(pic6,namud);

pic7= rgb2gray(imread('7.jpg')); pic7 = imresize(pic7,namud);

pic8= rgb2gray(imread('8.jpg')); pic8 = imresize(pic8,namud);

pic9= rgb2gray(imread('9.jpg')); pic9 = imresize(pic9,namud);

pic10=rgb2gray(imread('10.jpg'));pic10=imresize(pic10,namud);

[m,n] = size(pic1); %所取图片像素大小与1.jpg一致

% 下面采用一个细胞体结构的数据类型存储多个矩阵

pic\_all={pic1,pic2,pic3,pic4,pic5,pic6,pic7,pic8,pic9,pic10};

for i=1:10

%把m\*n的矩阵变换成1\*(m\*n)的矩阵

ImgData(i,:) = reshape(pic\_all{i},1,m\*n);

end

%讲数据范围缩小到0到1之间

ImgData = double(ImgData)/255;

**PCA.m：**

简单算法

输入：样本集合矩阵：img

降维的维数 ：k

输出：细胞结构体数据 ：Cell\_all

function Cell\_all = PCA(imgdata,k)

%reshape函数：改变句矩阵的大小，矩阵的总元素个数不能变

%img = [1,2;2,1;3,3;3,6;6,3];

% k = 2;

% img = double(imgdata);

[m,n] = size(imgdata); %取大小

img\_mean = mean(imgdata); %求每列平均值

img\_mean\_all = repmat(img\_mean,m,1);%复制m行平均值至整个矩阵

Z = imgdata - img\_mean\_all;

T = Z'\*Z;%协方差矩阵

[V,D] = eigs(T,k);%计算T中最大的前k个特征值与特征向量

img\_new = imgdata\*V\*D; %低维度下的各图片的数据

Cell\_all = {img\_new,V,D};

**find.m：**

人脸匹配

输入：细胞结构体数据Cell\_all（包括样本集合，特征值与特征向量）

想要识别的人脸（彩色图像）

输出：匹配的结果

function FaceFind = facefind(Cell\_all,img2find)

%细胞结构体的调用

img\_all = Cell\_all{1};

[m1,n1] = size(img\_all);

V = Cell\_all{2};

D = Cell\_all{3};

namud = 0.5; %图片缩小的倍数

%对需要识别的图像进行灰度等的处理

pic = rgb2gray(img2find); %灰度处理

pic = imresize(pic,namud); %变换大小

[m2,n2] = size(pic);

pic = reshape(pic,1,m2\*n2); %重新排列

pic = double(pic)/255;

pic\_done = pic\*V\*D; %处理完的数据

% 归一化，避免运算出现特别大的数据

Ma = max(max(pic\_done));

Mi = min(min(pic\_done));

pic\_done = pic\_done/(Ma - Mi);

for i=1:m1

% 归一化

Ma1 = max(img\_all(i,:));

Mi1 = min(img\_all(i,:));

img\_all(i,:) = img\_all(i,:)/(Ma1 - Mi1);

%求范数，把他们之间的几何距离作为评判与哪一个人脸最近的标准

error(i) = norm(img\_all(i,:)-pic\_done);

end

%找到其中最近的就认为是所要识别的人脸

Find = find(error == min(error));

% Find = error;