

实验二 静态图像分割

一、实验目的

- 1、使学生通过实验体会一些主要的分割算法对图像处理的效果，以及各种因素对分割效果的影响；
- 2、使用 Matlab 软件进行图像的分割；
- 3、能够自行评价各主要算子在有无干扰环境下的分割性能；
- 4、能够掌握分割条件(算法中各种参数等)的选择；
- 5、完成规定图像的处理并要求正确评价处理结果，能够在算法原理层面作出合理的解释。

二、实验内容

- 1、将图1-图5水表图像中的指示指针从背景中分离（指针部分显示为白色，其余部分显示为黑色）；（注：所有图采用统一代码）



图 1



图 2



图 3



图 4



图 5

- 2、利用二值形态学对分割后的二值图像进行处理以消除分割噪声；
- 3、根据分割结果给出水表读数（各位置指针所对应的单位可通过人为先验确定）。

三、关键方法（此处所提及的方法只是一个提示，并不需要全部用，更不是依次用，也不是唯一的方法）

1. 阈值分割法（也可以用其它分割算法）

基本原理是：通过设定不同的特征阈值，把图像像素点分为若干类。

常用的特征包括：直接来自原始图像的灰度或彩色特征；由原始灰度或彩色值变换得到的特征。

设待分割图像为 $f(x, y)$ (灰度图像或某颜色通道图像), 按照一定的准则 $f(x, y)$ 中找到阈值 T , 将图像分割为两个部分, 分割后的图像为: 若 $f(x, y) > T$, 则 $g(x, y) = 0$ (黑色), 若 $f(x, y) \leq T$, 则 $g(x, y) = 1$ (白), 即为我们通常所说的图像二值化。

Matlab 关键代码 (示例) 为:

```
I=imread('xl-130-16.tiff');  
f=rgb2gray(I); % 选取灰度通道, 也可尝试采用各彩色通道  
g=zeros(size(f));  
id = f <= T;  
g(id)=1;
```

2、基于区域的分割方法

包括区域生长法和分裂合并算法。算法原理见讲义。算法代码可参考冈萨雷斯《数字图像处理 (Matlab 版)》或网上资源。

3. 聚类分割算法

包括 k-means、EM 和 mean-shift 算法。各算法原理见讲义。算法代码可采用网上资源。

4. 数学形态学运算

经过阈值化后的二值图片并不完美 (即不仅药板部分为白色, 其它部分也存在白色斑点), 为此, 可采用二值形态学对其进行处理。

二值形态学中的运算对象是集合, 通常给出一个图像集合和一个结构元素集合, 利用结构元素对图像进行操作。其基本运算有四种: 腐蚀、膨胀、开运算和闭运算。基于这些基本运算和组合来进行图像形状和结构的分析及处理。如果 A 是图像集合, B 是结构元素 (B 本身也是一个图像集合), 形态学运算将使用 B 对 A 进行操作。结构元素往往比图像小得多。基本运算将遵循这个原则。

可能采用的二值形态学操作包括:

□ 膨胀和腐蚀

膨胀是在二值图像中“加长”或“变粗”的操作。这种特殊的方式和变粗的程度由一个称为结构元素的集合控制。

数学上, 膨胀定义为集合运算。 A 被 B 膨胀, 记为 $A \oplus B$

腐蚀“收缩”或“细化”二值图像中的对象。像在膨胀中一样, 收缩的方式和程度由一个结构元素控制。腐蚀的数学定义与膨胀相似, A 被 B 腐蚀记为 $A \ominus B$

数学形态学膨胀的matlab程序为:

```
se=strel('disk',3); % 可更改结构元素的大小  
A1=imdilate(A,se); % 膨胀 (A为二值图像)
```

数学形态学腐蚀的matlab程序为:

```
se=strel('disk',10); % 可更改结构元素的大小 (试着将10更改为5和20)
```

```
A2=imerode(A,se); % 腐蚀
```

□ 开运算和闭运算

在图像处理的实际应用中，更多地以各种组合的形式来使用膨胀和腐蚀，它们可以级连结合使用。膨胀后再腐蚀，或者腐蚀后再膨胀，通常不能恢复成原来图像（目标），而是产生一种新的形态变换，这就是开运算和闭运算。

A 被 B 的形态学开运算可以记做 $A \circ B$ ，这种运算是 A 被 B 腐蚀后再用 B 来膨胀腐蚀结果：

$$A \circ B = (A \ominus B) \oplus B$$

A 被 B 的形态学闭运算记做 $A \bullet B$ ，它是先膨胀再腐蚀的结果

$$A \bullet B = (A \oplus B) \ominus B$$

当处理二值图像时，采用上述的形态学变换组合，主要应用于提取某一区域的边界线、图像边缘轮廓、物体骨架特征和目标识别等众多的实际应用。

数学形态学开、闭的matlab程序为：

```
se=strel('square',20);  
fo=imopen(A,se);  
figure,imshow(fo);title('开运算后')  
fc=imclose(f,se);  
figure,imshow(fc);title('闭运算后')
```

□ 孔洞填充

若二值图像中存在孔洞需要被填充时，可采用 `imfill()` 命令，示例为：

```
BW2 = imfill(A, 'holes');
```

□ 连通域标记

若需要对二值图像中不同的目标点进行处理和分析时，需进行连通域标记，其程序如下：

```
L=bwlabel(A); % 连通域标记，A 为待标记二值图像
```

```
figure,map=[0 0 0;jet(10)];imshow(L+1,map);
```

5. 连通域分析

通过提取连通域的大小、形状、颜色、纹理等特征对目标区域进行判别。

6. 边缘检测

可利用微分算子进行边缘提取，主要的算子包括 Robert 算子，Sobel 算子，Prewitt 算子，LoG 算子和 Canny 算子等。Matlab 函数为：

```
edge(图像 [, edge 类型] [, 阈值])
```

比如 `edge(img, 'sobel');`

可选的 edge 类型有 'sobel', 'prewitt', 'log', 'roberts', 'canny' 等。

四、实验步骤

- 1 打开计算机，启动 MATLAB 程序；
- 2 调入数字图像，根据实验内容，参考关键方法，进行图像分割处理；
- 3 分析不同的分割方法或方法中的参数对结果的影响；
- 4 记录和整理实验报告。

五、实验仪器

- 1 计算机；
- 2 MATLAB 等程序；
- 3 移动式存储器（软盘、U 盘等），存储处理图像和子函数代码。
- 4 记录用的笔、纸。

六、实验报告内容

- 1 阐述实验过程，分析实验中出现的問題；
- 2 分析不同的方法或方法中的不同参数对实验结果的影响；
- 3 分析干扰环境下检测时结果上的差异；
- 4 阐述实验心得。

实验报告要求：纸张大小 A4，小四号字，1.25 倍行距，基本信息，正文（包括：实验目的，实验内容，实验步骤+源程序（加中文注释）+实验算法分析，实验结果（图形）+结果分析和讨论。

实验三 运动目标分割

一、实验目的

1. 采用差分法，背景差法实现运动目标提取，基本掌握运动目标提取的基本原理和方法
2. 了解不同帧率、不同运动速度条件下对检测结果的影响。

二、实验内容

- 1 利用MATLAB读取图像的基本命令imread和读取视频的基本命令aviread或mmreader;
- 2 调用各种运动目标检测函数进行运动目标分割;
- 2 利用MATLAB形态学算子调整分割结果;
- 3 了解形态学滤波中结构元strel命令的调用格式，通过改变结构元形状和大小，比较运动目标检测效果

三、关键方法

1. 帧差法

```
tic;
clear all
mov = mmreader('????.avi');
x = read(mov,2);
fr_size = size(x);
width = fr_size(2);
height = fr_size(1);
c=zeros(height, width);
for k=1:mov.NumberOfFrames-1
    x = read(mov,k);
    y = read(mov,k+1); % 可将1改为其它值，观察结果的不同
    m = rgb2gray(x);
    n = rgb2gray(y);
    q=im2double(n);
    w=im2double(m);
    c = q - w ;
    t=40; %% 阈值，此值可以调节
    t=t/255;%%转化为double型数据
    id = c > t;
    cc=zeros(fr_size);
    cc(id) = 1;
    figure(1),subplot(2,2,1),imshow(m)
    subplot(2,2,2),imshow(n)
```

```

subplot(2,2,3),imshow(c)
figure(2),imshow(cc) ;
Mov1(k) = im2frame(uint8(cc),gray);          % put frames into movie
end
movie2avi(Mov1,'fd_output','fps',15);

```

2. 背景差法

2.1 时间平均法

```

% This M-file uses average model for background pixels
clear all;
N = 40;% N frames to train the Gaussian model
h = fspecial('gaussian');
imagedata = filter2(h,rgb2gray(imread('****.bmp'))); % 根据实际图片序列信息进行修改
mu = imagedata;
[m,n] = size(mu);
pro = zeros(m,n);
for i=1:N
    filename = sprintf('*****/0%d.bmp',i); % 根据实际图片序列信息进行修改
    tmp = filter2(h,rgb2gray(imread(filename)));
    mu = mu+tmp;%(tmp+(i-1)*sav_mu)./i;%
end;
mu=mu./N;
figure(1),imshow(uint8(mu));
% test the new frame
for num = N+1:500
    filename = sprintf('*****/0%d.bmp',num);
    imagedata = filter2(h,rgb2gray(imread(filename)));
    t=20; % 阈值，可调节
    pro = abs(imagedata-mu)> t;
    imshow(pro),title(sprintf('frame number %d',floor(num)));
    %% update mean
    mu = (mu*(num-1) +imagedata)/num; %(1-a)*mu+a*(imagedata-mu);
    % display the results
    Mov1(num-N) = im2frame(uint8(255*pro),gray);          % put frames into movie
end;

```

```
movie2avi(Mov1,'ave_output_20','fps',15);
```

2.2 单高斯法

```
% This M-file uses single Gaussian model for background pixels
```

```
clear all;

N = 40;% N frames to train the Gaussian model

h = fspecial('gaussian');

imagedata = filter2(h,rgb2gray(imread('*****.bmp'))); %根据实际图片信息进行修改
mu = imagedata;
[m,n] = size(mu);
cov = zeros(m,n);
pro = zeros(m,n);
sav_mu = mu;
a = 0.01;
for i=1:N
    filename = sprintf('***%0%d.bmp',i); %根据实际图片信息进行修改
    tmp = filter2(h,rgb2gray(imread(filename)));
    mu = (tmp+(i-1)*sav_mu)./i;
    cov = ((tmp-mu).^2+(i-1)*cov)./(i)+(mu-sav_mu).^2;
    sav_mu = mu;
end;
cov = cov+0.001; %防止cov为0
% test the new frame
for num = N+1:500
    filename = sprintf('vipman_bmp%0%d.bmp',num);
    imagedata = double(filter2(h,rgb2gray(imread(filename))));
    T=1e-3; % 阈值，可调节
    pro = (2*pi)^(-1/2)*exp(-0.5*(imagedata-mu).^2./cov)./sqrt(cov)< T;
    %% update covariance and mean
    mu = mu + a*(1-pro).*(imagedata-mu); %(1-a)*mu+a*(imagedata-mu);
    cov = cov + a*(1-pro).*((imagedata-mu).^2-cov);%(1-a)*cov+a*(imagedata-mu).^2;
    % display the results
    imshow(mat2gray(pro)),title(sprintf('frame number %d',floor(num)));
    Mov1(num-N) = im2frame(uint8(255*pro),gray); % put frames into movie
    pause(0.000001);
```

end;

movie2avi(Mov1,'gaussian_output','fps',15);

2.3 混合高斯法（选做）

（自行查阅网上资料及代码）

四、实验步骤

- 1 打开计算机，启动 MATLAB 程序；
- 2 调入数字图像序列，根据实验内容，参考关键方法，进行运动目标分割处理；
- 3 分析不同的方法所对应的结果；
- 4 分析当阈值等参数发生改变时，对结果的影响；
- 3 记录和整理实验报告。

五、实验仪器

- 1 计算机；
- 2 MATLAB 等程序；
- 3 移动式存储器（软盘、U 盘等），存储处理图像和子函数代码。
- 4 记录用的笔、纸。

六、实验报告内容

- 1 阐述实验过程，分析实验中出现的問題；
- 2 分析不同的方法，不同参数条件对实验结果的影响；
- 3 阐述实验心得。

实验报告要求：纸张大小 A4，小四号字，1.25 倍行距，基本信息，正文（包括：实验目的，实验内容，实验步骤+源程序（加中文注释）+实验算法分析，实验结果（图形）+结果分析和讨论。

七. 拓展实验

1. 对检测出的车或人根据形状等简单特征进行判别，并利用 text 命令在图上做标注。

实验三 人脸检测

一、实验目的

理解和掌握人脸检测的基本理论和算法，练习使用肤色颜色空间建模+连通域处理及分析和 Harr-cascade 方法进行人脸检测；

二、实验内容

- 1 建立多种肤色模型，结合数学形态学滤波，完成人脸检测；
- 2 利用 Matlab 自带的计算机视觉系统工具箱实现单人及多人的人脸检测。

三、关键方法

1 肤色模型

基于肤色特征的人脸区域检测方法主要包括三个步骤：

- (1) 将图像映射到选定的颜色空间；
- (2) 在选定的颜色空间中，依据肤色模型逐个判定像素色彩是否为肤色；
- (3) 将图像中的肤色像素按照区域聚类，进行连通域分析，完成人脸区域定位。

1.1 HSV 颜色空间及肤色模型

HSV 颜色空间由分别代表色调、饱和度、亮度的成分组成，属于非线性颜色空间。在 HSV 颜色空间中，用色度和饱和度即可描述人脸肤色信息。色度-饱和度信息通过专门的灰度饱和度和度图进行肤色分割，如图 3-1 所示，这个一个从六角锥中截出的六边形，阴影区域为肤色在 HS 子空间中的分布示意图，肤色在 HS 空间中的分布为环形的一部分。

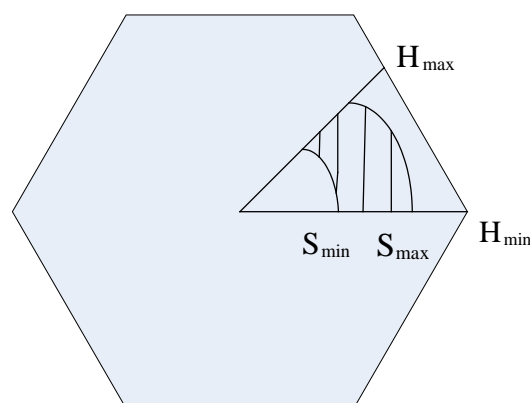


图 3-1 肤色在 HS 子空间的分布

图 3-1 说明人脸肤色在 HS 子空间的分布情况。在一定光照条件下，人脸肤色的聚类结果是

$S_{\min} = 0.23, S_{\max} = 0.68, H_{\min} = 0, H_{\max} = 50$ 。

参考代码：

```
files=dir('* .jpg');  
for i=1:length(files)  
    s=files(i).name;  
    I=imread(s);
```

```

I1=rgb2hsv(I);
h=I1(:,:,1);
s=I1(:,:,2);
id = s > 0.23 & s < 0.68 & h > 0 & h < 50;
figure,subplot(2,1,1),imshow(I),
subplot(2,1,2),imshow(id)
end

```

1.2 YCbCr 颜色空间及肤色模型

YCbCr 颜色空间也是一种常见的色彩空间，经实验，该颜色空间可以较好地描述具有高斯分布的人脸肤色，依据该空间建立的肤色模型可以描述如下：

令 $\mathbf{x} = (Cb, Cr)^T$ 表示某像素点的肤色值，则该像素点的肤色相似度可以表示为：

$$p(\mathbf{x}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{m})\right]$$

其中

$$\mathbf{m} = (Cb, Cr)^T,$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{Cb,Cb} & \sigma_{Cb,Cr} \\ \sigma_{Cr,Cb} & \sigma_{Cr,Cr} \end{bmatrix}$$

通过大量样本训练，肤色均值和方差参数为： $Cb = 117.44$ ， $Cr = 157.56$ ， $\sigma_{Cb,Cb} = 299.46$ ，

$\sigma_{Cb,Cr} = 12.14$ ， $\sigma_{Cr,Cb} = 12.14$ ， $\sigma_{Cr,Cr} = 160.13$ 。代入这些参数，计算每个像素点对应的肤色相似度，

相似度大于设定阈值的像素点即可判定为肤色像素点。

```

files=dir('*.jpg');
for i=1:length(files)
    s=files(i).name;
    I=imread(s);
    I=imresize(I,0.1);
    I1=rgb2ycbcr(I);
    I1=double(I1);
    cb=I1(:,:,2);
    cr=I1(:,:,3);
    mu=[117.44 157.56]';
    sig=[299.46 12.14;12.14 160.13];
    p=drawGaussian(mu,sig,cb(:),cr(:));

```

```

p=reshape(p,size(cb));
figure(1),subplot(2,2,1),imshow(I),
subplot(2,2,2),imshow(p)
p1 = p > 0.5;
subplot(2,2,3),imshow(p1);
end

```

1.3 连通域分析

利用上述方法得到的目标点存在一些噪声点，可利用形态学方法进行处理，并用 `bwlabel` 函数进行标记，因为人脸在图像中尺寸较大，且内部连续，类似椭圆，我们可以根据外接矩形的长宽比，人脸矩形区域二值图像占空比滤除一些孤立小目标或者形状与人脸差异较大的目标，得到较为准确的人脸区域定位结果。

可能用到的函数包括：

```

bwopen
bwclose
bwareaopen
bwlabel 等。

```

代码自行编写。

2 Harr-cascade 人脸检测

主流的人脸检测方法采用 Adaboost 学习算法，Adaboost 算法是一种用来分类的方法，它把一些比较弱的分类方法合在一起，组合出新的很强的分类方法。

人脸检测过程中使用 Adaboost 算法挑选出一些最能代表人脸的矩形特征(弱分类器)，按照加权投票的方式将弱分类器构造为一个强分类器，再将训练得到的若干强分类器串联组成一个级联结构的层叠分类器，有效地提高分类器的检测速度。

MATLAB 的 Computer Vision System Toolbox （计算机视觉工具箱）已经实现（或者说自带）人脸检测功能，代码如下：

```

% Create a cascade detector object.
faceDetector = vision.CascadeObjectDetector();

Read an image
I = imread('face.jpg');
I=im2double(I);
bbox = step(faceDetector, I);

% Draw the returned bounding box around the detected face.

```

```
boxInserter = vision.ShapeInserter('BorderColor','Custom','CustomBorderColor',[255 255 0]);  
J = step(boxInserter, I, bbox);  
figure, imshow(J), title('Detected face');
```

大家可将参考上述代码，将输入换为视频数据的获取。

四、实验步骤

- 1 打开计算机，安装和启动 MATLAB 程序；
- 2 调用数字图像，参考实验指导书，进行人脸检测操作；
- 3 对多幅图像和视频结果进行分析，了解各个方法的优缺点以及方法中参数对结果的影响。
- 4 记录和整理实验报告。

五、实验仪器

- 1 计算机及操作系统。Windows XP 或以上版本。
- 2 软件。Matlab2012b 及以上版本。
- 3 彩色人脸图像。
- 4 移动式存储器（软盘、U 盘等），存储处理图像。
- 5 记录用的笔、纸。

六、实验报告内容

- 1 叙述实验过程，分析实验中出现的問題；
- 2 分析不同方法的优缺点；
- 3 分析方法中参数的变化对结果产生的影响。
- 4 撰写实验心得。

七、拓展实验

- 1 利用 matlab 自带工具箱，结合脸部检测结果，检测眼睛、嘴巴和鼻子，并思考如何删除误检测。

实验四 人脸识别

一、实验目的：

- 1、学会使用 PCA 主成分分析法。
- 2、初步了解人脸识别的特征法。
- 3、更熟练地掌握 matlab 的使用。

二、实验原理：

1、PCA（主成分分析法介绍）

PCA 方法的基本原理是：利用离散 K-L 变换提取人脸的主要成分，构成特征脸空间，识别时把测试样本投影到该空间，构成一组投影系数，通过与特征脸的距离比较，距离最小的特征脸对应的即是识别结果。

基于 PCA 的人脸识别分为三个阶段，第一个阶段利用训练样本集构建特征脸空间；第二个阶段是训练阶段，主要是将训练图像投影到特征脸子空间上；第三个阶段是识别阶段，将测试样本集投影到特征脸子空间，然后与投影后的训练图像相比较，距离最小的为识别结果。基于 PCA 的人脸识别其实一种统计性的模板匹配方法，原理简单，易于实现，但也有不足，它的识别率会随着关照，人脸角度，训练样本集的数量而变换，但仍不失为一种比较好的方法。

三、实验步骤

1 打开计算机，安装和启动 MATLAB 程序；

2 将库里的 400 张照片（40 个个体，每个个体 10 张照片）分成两组。一组作为训练，一组作为测试。可尝试下列分组情况：

(1) 每个人的任意 5/8 张照片作为训练并作为测试样本库，其它 5/2 张作为测试待识别图片。（即测试时所识别的个体出现在训练库中，后面的步骤描述和参考代码以此为基础）

(2) 前 30/38 个人作为训练，后 10/2 个人作为测试，其中测试库中每个人的任意 5/8 张照片为测试样本库，其它 5/2 张照片作为待识别图片。（即测试时所识别个体未出现在识别代码会与参考代码不同）

训练的照片按照顺序的数字重命名。

参考代码：

```
clear;
n = 1;
p=1;
for i=1:40
    a=1:10;
    Ind = a(:,randperm(size(a,2)));
    for h = 1:5
        j= Ind(1,h);
        File=['C:\Users\dell\Desktop\ORL\s',sprintf('%d',i),'\',sprintf('%d',j),'.pgm'];
        Filesave=['C:\Users\dell\Desktop\new',sprintf('%d',1),'\',sprintf('%03d',n),'.pgm'];
```

```

        copyfile(File,Filesave)
        n = n + 1;
    end

    for h = 6:10
        j= Ind(1,h);
        File=['C:\Users\dell\Desktop\ORL\s',sprintf('%d',i),'\',sprintf('%d',j),'.pgm'];
        Filesave=['C:\Users\dell\Desktop\new',sprintf('%d',2),'\',sprintf('%03d',p),'.pgm'];
        copyfile(File,Filesave)
        p = p + 1;
    end
end
end

```

3 库照片处理

① 将每一张库的照片转化成 N 维的向量，然后把这 200 个向量以列的形式存在了矩阵里。即

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{200}]$$

② 将这 200 个向量的每个元素相加起来求出平均值。再用 \mathbf{X} 里的每一个向量减去这个平均值得到每个的偏差。

$$\text{平均值 } \bar{\mathbf{x}} = \frac{1}{200} \sum_{i=1}^{200} \mathbf{x}_i, \quad \text{每个向量的偏差 } \mathbf{x}' = \mathbf{x} - \bar{\mathbf{x}}$$

最后得到

$$\mathbf{X}' = [\mathbf{x}'_1 \quad \mathbf{x}'_2 \quad \cdots \quad \mathbf{x}'_{200}]。$$

③ 计算协方差矩阵

参考代码如下：

```

% 批量读取指定文件夹下的图片
path = uigetdir;
img_path = dir(strcat(path, '*.*pgm'));
img_num = length(img_path);
imagedata = [];
if img_num > 0
    for j = 1:img_num
        img_name = img_path(j).name;
        temp = imread(strcat(path, '/', img_name));
        temp = double(temp(:));
        imagedata = [imagedata, temp];
    end
end

```

```

end
wts = size(imagedata,2);

% 中心化并计算协方差矩阵
img_pj = mean(imagedata,2);
for i = 1:wts
    imagedata(:,i) = imagedata(:,i) - img_pj;
end
covMat = imagedata'*imagedata;

```

④ 利用协方差矩阵进行主成分分析，提取特征脸，获取训练样本特征数据

参考代码如下：

%说明：[PC,latent,explained]=pcacov(X)通过协方差矩阵 X 进行主成分分析，返回主成分(PC)、%协方差矩阵 X 的特征值 (latent)和每个特征向量表征在观测量总方差中所占的百分数%(explained)。

```
[COEFF, latent, explained] = pcacov(covMat);
```

% 选择构成 95%能量的特征值

```
i = 1;
```

```
proportion = 0;
```

```
while(proportion < 95)
```

```
    proportion = proportion + explained(i);
```

```
    i = i+1;
```

```
end
```

```
k = i - 1;
```

%求出原协方差矩阵的特征向量，即特征脸

```
V = imagedata*COEFF; % N*M 阶
```

```
V = V(:,1:k);
```

% 训练样本在 PCA 特征空间下的表达矩阵 k*M

```
W = V'*imagedata;
```

```
msgbox(['训练完成'])
```

⑤ 对于测试样本，读取图像，将其变换至 PCA 空间，利用近邻法识别

参考代码：

```
im=imread('test_face.jpg');
```

```
im = double(im(:));
```

```
objectone = V'*(im - imgpj);
```

```
distance = 1e8;
for k = 1:wt
    temp(k) = norm(objectone - W(:,k));
end
[s_temp,id]=sort(temp,'ascend');
```

⑥ 显示识别结果(为便于分析，对每一幅测试图像显示距离排名在前三位的识别结果)

% 代码自编

5 记录和整理实验报告。

四、实验仪器

1 计算机；

2 MATLAB 程序；

3 移动式存储器（软盘、U 盘等），存储处理图像。

4 记录用的笔、纸。

五、实验报告内容

1 叙述实验过程，分析实验中出现的问題，并显示前 5 张特征脸；

2 记录不同数量训练样本条件下人脸识别的准确率，并以图表形式呈现；

3 对实验结果进行分析；

4 进行实验总结，写心得。

六、拓展实验

1 利用 Matlab GUI 设计一个简单的人脸识别软件。