介绍

背景

特征脸是关于人脸识别的计算机视觉的关键概念。根据Matthew和Alex \ cite {1}的说明，实际的特征面是一组特征向量，用于表示具有相应适当权重值的图像。这种机制是极端LY 与利用笛卡尔协调系统来描述三维空间中的点相似。后者利用了在x，y和z轴，以确定的位置的点和前利用特征脸，以确定一个图像。在这个过程中，一组本征面 可以称为识别图像的一组描述基础。

介绍

为了初步了解计算机人脸识别的基本原理并在此过程中发挥本征人脸的作用，本实验分为六个部分，并使用软件\ emph { matlab }。在练习之前，需要进行一些必要的操作。要下载文件名为\ EMPH {数据\ \_for \ \_ labC.mat } 中的\ EMPH { Matlab的}，然后检查变量\ EMPH { eigenfaces\_blk }和\ EMPH { emplo yees\_DB }。前者包含101个参数，并且如果它们以图像形式显示，则前100个参数是特征脸，这是因为它们具有类似面孔的外观。在本实验中，变量\ emph { eigenfaces\_blk } 的最后一个参数实际上是多余的，下面将在后面的文章中进行解释。变量\ emph { emplo yees\_DB }是一个结构，其中包含100名员工的ID号和描述这些员工的面孔的相应权重。每个部分的任务如下所示。

1. 任务1：验证变量\ emph { eigenfaces\_blk } 中的这些本征面是否正交。
2. 任务2：使用给定的egienfaces ，将输入图像投影到egienfaces 的整个集合（训练脸）上，即可找到输入图像的权重。由于投影在每个本征面上的输入图像将生成一个权重值，因此将获得一组与本征面数量相同大小的权重。需要编写\ emph { Matlab }函数来获得面部的权重。函数的声明应类似于：

$ Function [weight \ \_of \ \_face，mean of\_face ] = get \ \_face \ \_weights（im，eigenfaces\_blk ）$

                其中\ emph {mean \ \_of \ \_face}是图像的平均强度\ emph { im }。使用此功能获得图像的权重\ emph { find \ \_id.jpg}。然后绘制权重并评论该图是否传达了员工的任何信息。

1. 任务3：编写一个\ emph { Matlab }函数，使用从上一个任务获得的图像\ emph {find \ \_id.jpg} 的权重，生成相应员工的图像。此函数的声明应类似于：

$ function [ im ] = 从\ \_weights生成\ \_face \ \_（weights \ \_of \ \_face，eigenfaces\_blk ，均值\ \_of \ \_face）$

最后，说明为什么100个特征脸足以描述具有450×300像素的脸的图像。

1. Taks4：编写一个\ emph { Matlab }函数，通过其图像查找员工的ID号，以实现人脸识别的目的。此任务的主要思想分为两个步骤。首先是通过使用函数\ emph { get \ \_face \ \_weights } 获得输入图像的权重。第二步是比较输入图像的权重与数据库中不同员工的权重之间的欧几里得距离。当距离最小时，可以认为数据库中的相应权重和输入图像的权重属于同一张人脸图像。函数的声明应类似于：

$ function [ID ] = get \ \_employees \ \_ID \ \_from \ \_DB（im ，employee\_DB ，...）$

然后利用此功能获得图像的ID号\ emph { find \ \_id \ \_jpg}。

1. 任务5：通过使用先前任务中的函数来访问人脸识别的鲁棒性。对于同一个人，由于不同的条件和仅用于描述特定图像的一组权重，因此可以获得不同的面部图像。因此，输入图像可能与数据库中权重描述的图像不同。函数\ emph { get \ \_employees \ \_ID \ \_from \ \_DB }是否仍然可以识别图像并找到需要检查的员工的ID号。在此任务中，在原始图像\ emph { find \ \_id.jpg} 上添加了两种类型的噪声\ emph {AWGN}和\ emph {盐\＆pepper }。\ mph {AWGN} 的五个标准偏差的识别结果需要进行比较，并且还需要测试\ mph {salt \＆pepper}噪声的五个噪声密度的识别结果。
2. 讨论具有较大特征脸空间的优缺点，并总结有关基于特征脸的图像识别算法的鲁棒性的发现。

方法论与结果

任务1：

为了验证特征面是正交的，可以激发图像变换的知识以应用于该任务。正交变换基的系数根据以下显示的算法获得。

在此，特征面是描述图像的基础，权重是上图中的系数。因此，权重的值可以通过以下等式s找到：

     面（j）= \ sum\_ { i = 1} ^ 101 employee\_DB（j）。weights（i）。\* eignfaces\_blk （：，：，i ）

其中变量$ j $是员工人数，$ i $是权重值101的数量。\ textbf { 如果特征面如上图所示是正交的，则将满足以下方程式} 。

      employee\_DB（j）。weights（i ）=平均值{face（j）。\* eignfaces\_blk （：，：，i ）}

因此，可以利用该方程式来验证这些本征面的正交化。\ emph { Matlab }代码如下所示。

清除;

加载data\_for\_labC.mat ;

对于j = 1：1：100

    a = 零（450,300）;

对于我= 1：1：101 ％ -找到的图像数据“A”每位员工的

    A = A + employees\_ DB （j）的.weights（我）\* eignfaces\_blk （：，：，我）;

结束

因为这些egienface 是正交的

b = a。\* eignfaces\_ blk （：，：，1）;

实际上，c（j）是数据库中第j个权重的第一个值

c（j）= round（平均值（平均值（b）），4）;

d（j）= round（employee\_DB （j）.weights（1），4）;

结束

结果= 等值（c，d ）

在方程\ ref { }中，$ i $ 的值被选择为1，因此，矩阵\ emph {c}将保持与数据库中所有权重的第一个值相同的值。为了方便比较，提取数据库中所有权重的第一个值并将其放入矩阵\ emph { d}中。如果矩阵\ emph {c}和\ emph {d}相等，则方程ref { }将成立。

运行\ emph { Matlab }代码，命令窗口显示如下：

由于函数\ emph { 等值}返回1，表示矩阵\ emph {c}和emph { d}相等。则方程\ ref { }是正确的，特征面是正交的。

任务2：

功能\ EMPH {GET \ \_face \ \_weights}被写入通过应用方程\ REF { } 如下：

函数[weights\_of\_face，mean\_of\_face] = get\_face\_weights（im，eigenfaces\_blk）

weights\_of\_face = zeros（1,101）;

im = double（im ）;

对于我= 1：1：101

    mean\_of\_face = 平均值（mean（im ））;

    weights\_of\_face （i ）=平均值（mean（im 。\* eigenfaces\_blk （：，：，i ））））;

结束

该任务的\ emph { Matlab }代码如下所示：

清除;

im =未读（'find\_id.jpg' ）;

加载data\_for\_labC.mat ;

eigenfaces\_blk = eignfaces\_blk ;

[weights\_of\_face ，mean \_of\_face] = get\_face\_ weights（im，eigenfaces\_blk）;

图;

情节（weights\_of\_face ）;

title（'输入图像的权重' ）;

ylabel （'Weights values' ）;

xlabel （'Number' ）;

格上;

在\ emph { Matlab } 的\ emph {Workspace}中可以找到运行图像的代码和权重\ emph {find \ \_id.jpg }。部分屏幕截图如下所示（由于纸张的宽度有限）。

图像\ emph {find \ \_id.jpg} 的权重如下所示。

从图中\ ref {}来看，与第1、3 和101号本征面相对应的权重值大于相应本征面的其他权重值，这表明输入图像包含第1、3和101号本征面的更多详细信息。

任务3：

\ emph { Matlab }函数\ emph { generate \ \_face \ \_from \ \_weights }是使用公式\ ref { } 编写的。

函数[ im] = generate\_face\_from\_weights（weights\_of\_face，eigenfaces\_blk，mean\_of\_face ）

im = 零（450,300）;

对于我= 1：1：101

    im = im + weights\_of\_face （i ）。\* eigenfaces\_blk （：，：，i ）;

结束

im = uint8（im ）;

该任务的代码如下所示。

清除;

im =未读（'find\_id.jpg' ）;

加载data\_for\_labC.mat ;

eigenfaces\_blk = eignfaces\_blk ;

[weights\_of\_face ，mean \_of\_face] = get\_face\_ weights（im，eigenfaces\_blk）;

imre = generate\_face\_from\_weights （weights\_of\_face，eigenfaces\_blk ，mean\_of\_face ）;

子图（121）;

imshow （im ）;

title（'原始图像​​' ）;

子图（122）;

imshow （imre ）;

title（'按权重重建的图像' ）;

PSNR = calculate\_psnr （IM，伊姆雷）;

运行代码，并获得原始输入图像和重建图像，如下所示。

当然，通过使用度量\ emph { PSNR} 可以测量重建图像的质量。所述\ EMPH 原始图像和重建图像之间{PSNR}值在\发现EMPH 如下所示{工作区}。

\ emph {PSNR}值是无限的，因此，重建图像的质量非常大。

根据引文\ cite {1}，训练图像的大小为450×300像素，这说明本征面空间应为450×300尺寸。尽管如此，本文还是介绍了另一种简化的构造本征面空间的替代解决方案。如果训练图像的数量M 小于训练图像的大小N \ N ，则$ M-1 $ 特征脸足以构造脸部图像。在此任务中，所有训练图像的训练图像数为100，大小为450 \ x300。\ textbf { 因此，99个特征面足以描述图像。当然，100个特征脸也是可以描述图像的。使用100个特征脸来描述图像的缺点是描述的基础不是最简单的，并且会产生冗余。}

任务4：

在此任务中，应用了欧几里得距离的概念。为了在数据库中找到与输入图像\ emph {find \ \_id.jpg} 的权重相差最小的权重，利用欧几里得距离来描述两个权重之间的差，公式如下。

当然，此处使用之前编写的函数\ emph { get \ \_face \ \_weights }来获取输入图像的权重。函数\ emph { get \ \_employees \ \_ID \ \_from \ \_DB }编写如下。

函数[ID] = get\_employees\_ID\_from\_DB （im ，employee\_DB，eigenfaces\_blk ）

[weights\_of\_face ，mean \_of\_face] = get\_face\_ weights（im，eigenfaces\_blk）;

a = 规范（weights\_of\_face-employees\_DB （1）.weights）; ％-假设最近的距离

对于我= 1：1：100

    b = 规范（weights\_of\_face-employees\_DB （i ）.weights）;

    c = b;

    如果（a> = c）％-更新最近的距离

        a = c;

        j = i ;

    结束

结束

ID = 员工\_ DB （j）.id;

\ emph { Matlab }脚本写为打击：

清除;

加载data\_for\_labC.mat ;

im =未读（'find\_id.jpg' ）;

eigenfaces\_blk = eignfaces\_blk ;

ID = get\_employees\_ID\_from\_DB （im ，employee\_DB ，eigenfaces \_blk ）

运行代码，结果显示在\ rfe { } 图中显示的命令窗口中。

\ textbf {因此，图片的员工ID \ emph {find \ \_id.jpg}为96。}

任务5：

在此任务中，主要思想是在原始图像\ emph {find \ \_id.jpg} 上添加不同的噪点，然后利用任务4中编写的\ emph { get \ \_employees \ \_ID \ \_from \ \_DB } 函数来识别嘈杂的图像。如果ID号正确，则识别算法仍具有识别图像的强大功能。否则，识别算法将无效。

一种）。对于\ emph {AWGN}，将标准偏差选择为5、10、60、100 和150，以构造5个噪点图像。\ emph { Matlab }如下所示：

清除;

im =未读（'find\_id.jpg' ）;

加载data\_for\_labC.mat ;

eigenfaces\_blk = eignfaces\_blk ;

sigma = [5 20 60 100 150];

对于我= 1：1：5

im\_noise = wgn （450,300，sigma（i ）。^ 2）;

im1 = uint8（double（im ）+ double（im\_noise ））;

id（i ）= get\_employees\_ID\_from\_DB （im1，employee\_DB，eigenfaces\_blk ）;

结束

运行代码后，在\ emph { 工作区}的\ emph {id} 矩阵中找到五个嘈杂图像的识别ID号。

显然，可以正确地识别出具有标准偏差5的第一噪声图像的ID号96 ，而不能识别另一噪声图像。表\ ref { } 显示已识别的情况。

                                                                                 表格1

b）。对于\ emph { 盐\＆pepper } 噪声，将噪声密度选择为0.01、0.2、0.5、0.7 和0.9 以形成五种不同的噪声\ emph { Matlab }如下所示：

清除;

im =未读（'find\_id.jpg' ）;

加载data\_for\_labC.mat ;

eigenfaces\_blk = eignfaces\_blk ;

百分比＝ [0.01 0.2 0.5 0.7 0.9]；

对于我= 1：1：5

im1 =无噪声（im ，“盐和胡椒粉” ，百分比（i ））；

id（i ）= get\_employees\_ID\_from\_DB （im1，employee\_DB，eigenfaces\_blk ）;

结束

运行代码后，可以在\ emph {Workspace}中找到带噪图像的ID号，如下所示。

显而易见，当\ emph {salt \＆pepper} 的噪声密度为0.7和0.9时，识别结果75是不正确的。当噪声密度为0.01、0.3和0.5时，可以找到正确的ID号96，可以识别出嘈杂的图像。表\ ref { } 显示已识别的情况。

                                                                                 表2

任务6：

如果本征面空间较大，则将存在一些优点和缺点：

1）。优点

当特征脸空间变大，例如，从100 to10000，那里需要更多的训练图像来构造本征脸空间，这将导致\ textbf 由{更面部图像可以被描述和识别} 特征脸的空间。

2）缺点

一种）。较大的特征脸空间装置需要有\ textbf {更大的存储器}来存储更多的特征脸的计算机。

b）。由于权重的大小与特征面的数量相同，因此较大的特征面空间将导致权重的较大。因此，对计算机高速缓存的要求更高。

C）。的较大的特征脸空间和重量的较大尺寸将导致计算速度要慢，花费更多的时间来识别面部图像和效率会变得更低。

基于特征脸的图像识别算法的鲁棒性发现：

1）。对于\ emph {AWGN}，该算法实际上具有\ textbf { 阈值}，以标识比较的图像是否属于一个人。当欧几里德距离小于该阈值时，比较的图像属于同一个人。当欧几里得距离大于此阈值时，该算法将产生错误的标识并获得错误的结果。在此任务中，此阈值可以用\ emph {AWGN} 的标准偏差表示，并且位于标准偏差的5到20 的间隔内。

2）。作为用于\ EMPH {盐\＆胡椒}噪声，该阈值可以通过噪声的噪声密度来表示，并且位于在0.5的时间间隔中的噪声密度的0.7。

@ARTICLE {6793549，  
作者= {M。Turk and A. Pentland }，  
journal = {Journal of Cognitive Neuroscience}，  
书名= { Eigenfaces for Recognition}，  
年份= {1991}，  
卷= {3}，  
数字= {1}，  
页数{{71-86}，  
doi = {10.1162 / jocn.1991.3.1.71}，  
ISSN = {0898-929X}，  
month = {Jan}，}