统计分析方法作业3

16337183 孟衍璋

1 实验目的

使用PCA进行图像压缩。

2 实验要求

输入一张灰度图片Lena，放大到$256 * 256$，使用PCA方法把原始图片分别按照2 : 1、8 : 1、32 : 1进行压缩，即压缩后的数据量为原始图片的1/2、1/8、1/32。分析压缩后的数据所含信息量大小，并比较压缩数据再经过重建后与原始图片的视觉差异。

把图像分割成很多块$16 * 16$，把每个小图像块看成不同的样本点，一个小图像块内每个像素是样本点的不同维度。

3 实验步骤

首先，定义一个*compression*函数。输入为图像和压缩的比例，输出信息量与重构后的图。

实验要求里写到要先将图片分成$16 * 16$的小块，由于图像的像素为$512 * 512$，所以相当于要分成$32 * 32 = 1024$个小块。

这部分的代码如下：

```
%将图像分为16*16的小块，对于512*512的图片，一共32*32=1024块
X = zeros(16,16,1024);  %随机变量X1,X2,...,X1024
region_size=16;
numRow=32;
numCol=32;
t1 = (0:numRow-1)*region_size + 1; t2 = (1:numRow)*region_size;
t3 = (0:numCol-1)*region_size + 1; t4 = (1:numCol)*region_size;
%figure;
k = 0;
for i = 1 : numRow
    for j = 1 : numCol
        temp = image(t1(i):t2(i), t3(j):t4(j), :);
        k = k + 1;
        X(:,:,k) = temp;
%         subplot(numRow, numCol, k);
%         imshow(temp);
    end
end
```

这样便将每一小块图形存入了变量X中。

然后就开始分别对每一小块进行主成分分析，并使用对应的主成分重构图像：

```
%对每一部分的图像做主成分分析
Y = zeros(16,16,1024);
total = 0;
part = 0;
for i = 1:1024
    [u,s,v] = svd(double(X(:,:,i)));

    %重构压缩后的图像
    r = 2;    %压缩率
    %K = round(16 / r);
    K =round(2 * region_size * region_size / ( r * (region_size + region_size + 1)));

    if K > region_size
        K = region_size;
    end

    Y(:,:,i) = zeros(size(X(:,:,i)));
    for j = 1:K
        Y(:,:,i) = Y(:,:,i) + s(j,j) * u(:,j) * v(:,j)';    % 利用前K个特征值重构原图像
    end

    %计算信息量
    total = total + sum(diag(s));
    for w = 1:K
        part = part + s(w,w);
    end
end
```

使用PCA进行图像压缩并重构之后，还需要将之前的小块合成为一张图，代码实现如下：

```
%将小块合成一张图
image_compression = zeros(1,512);
for j = 1:32
    image_compression_row = Y(:,:,(j-1)*32+1);
    for k = 2:32
            image_compression_row = cat(2, image_compression_row, Y(:,:,(j-1)*32+k));
    end
    image_compression = cat(1, image_compression, image_compression_row);
end
image_compression(1,:) = [];
```

之后想要在$matlab$中显示图像的时候遇到了一个问题。输出的图像总是白色的，而且使用函数$imwrite()$存储得到的图片也是白色的。经过查询之后才得知，原来是之前读取的图像被转换为$double$类型，这时如果直接用$imwrite()$保存图片的话，就会出现全是白色的情况，因为在转化图像类型的时候，图像的像素值虽然变为$double$数据类型，但是取值范围仍然是$0-255$，$imwrite()$处理$double$类型的时候期待的数据范围是$0-1$，会把大于1的值当作1处理，这样几乎所有的像素都变成了白色。

所以这个时候就要先用函数$uint8()$转化数据类型，这样输出的图片就没有问题了。

最后就还需要计算信息量，其中压缩率是由代码中的参数r决定的，计算出的信息量如下：


原始图像
压缩为0.5 信息量:0.98207
压缩为0.25 信息量:0.94729
压缩为0.125 信息量:0.90308