

## 统计分析方法实验三：使用PCA进行图像压缩

信息安全 16337028 陈谔希

>

### 一.实验目的

利用PCA（主成分分析）进行图像压缩，进一步了解和掌握主成分分析的方法

### 二.实验内容

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。

### 三.实验及算法原理

我是使用matlab语言来做的这一次实验

#### 读取图片

首先我利用imread函数来读取原始的图片文件，随后建立变量记录原图片的尺寸以及建立矩阵记录原图片的灰度值，并且记录下所需要压缩的维度等数值，随后利用im2col函数对灰度矩阵进行分块，把这个512\*512的矩阵变为256\*1024的矩阵

```
clc;%清屏

%读取图片
img = imread('原始图片.bmp');

figure(1),subplot(232),imshow(img,[]);title('原始图片');%显示原图片

[M N] = size(img);%记录原图片的尺寸 在这里此图片是512*512

f = double(img);%保存原图片的灰度矩阵 512*512

bs = 16; %图像块尺寸

p1 = 128; %保留的维数
p2 = 32;
p3 = 8;

g = im2col(f, [bs bs], 'distinct'); %将图像块分为bs*bs的块，随后每一块重排列生成新矩阵
%以上把图片分成了N个不重叠的16*16的小块，也就是变成了N*256的数据集
%因此这里由512*512变为 (16*16) *1024 即256行*1024列 详见
https://www.cnblogs.com/rong86/p/3557193.html
```

#### 图像压缩

首先计算分块后每一块的灰度均值，随后之前分好块的图像灰度矩阵减去均值矩阵

```

g_m = ones(size(g,1),1)*mean(g);    %计算每块的灰度均值
%ones (创建一个新的矩阵 行数与g相同, 即256行, 每一行的值都为1)
%mean(g)输出每列的灰度均值, 共1024列 两者相乘得到256*1024矩阵, 每一列的256个元素都是mean(g)的值

g = g - g_m;%每个灰度减去该列的灰度均值

```

接下来就可以进行主成分分析了

首先求出图像灰度矩阵的协方差矩阵, 并利用eig函数来求出特征值的对角矩阵以及矩阵E

再用diag函数取出特征值, 并且将其排序再重新放回矩阵中

最后对排列好的E矩阵进行降维 (取最大的p个, p为取的维数),

```

covarianceMatrix = X*X'/(size(X,2)-1);%x乘x的转置 除以x矩阵的列-1 即1023 得出协方差矩阵

[E, D] = eig(covarianceMatrix); %返回特征值的对角矩阵 D 和矩阵 E, 其列是对应的右特征向量, 使得
covarianceMatrix*E = E*D。

[d_out,order] = sort(diag(D),'descend');%diag函数 取对角元素 即矩阵D中特征向量 sort 进行降序排列
与序号一起组成256*2的矩阵

E = E(:,order);%把E的列倒换, 即原来第x列变为第257-x列

D = diag(d_out);%把排列好的特征向量再放回到矩阵中去, 成为排列好的对角特征向量矩阵D

E_proj1 = E(:,1:p1);    %取最大的p个特征值所对应的特征矢量进行降维
E_proj2 = E(:,1:p2);
E_proj3 = E(:,1:p3);

g_proj1 = g'*E_proj1;    %从bs*bs维映射到p维 1024*256矩阵与256*p矩阵进行相乘, 最后得出1024*p矩阵
达到降维目的
g_proj2 = g'*E_proj2;
g_proj3 = g'*E_proj3;

```

## 图像复原

随后对降维的矩阵进行映射复原, 每个块再加上灰度均值, 最后再使用col2im函数把矩阵的每一列变为块重新组成图片

% 恢复图像

```
g_rec1 = g_proj1*E_proj1';%使用1024*p矩阵与256*p矩阵的转置（即p*256矩阵）相乘，得到一个1024*256的矩阵g_rec  
g_rec2 = g_proj2*E_proj2';  
g_rec3 = g_proj3*E_proj3';
```

```
s1 = g_rec1' + g_m;%加处理前每个块的灰度均值
```

```
s1 = col2im(s1, [bs bs], [M N], 'distinct');%将图像每一列变为bs*bs的块，随后每一块重排列生成新矩阵
```

%因此这里由256\*1024变为 (16\*16) \*2\*512 即512行\*512列

```
s2 = g_rec2' + g_m;
```

```
s2 = col2im(s2, [bs bs], [M N], 'distinct');
```

```
s3 = g_rec3' + g_m;
```

```
s3 = col2im(s3, [bs bs], [M N], 'distinct');
```

最后再计算图片信息量

%计算图片信息量 由于var方差函数无法直接计算矩阵方差，因此要先通过std标准差函数计算再平方随后矩阵求和求出所需要的数据

```
sum(sum((std(s1,0,1).^2)))/sum(sum((std(f,0,1).^2)))
```

```
sum(sum((std(s2,0,1).^2)))/sum(sum((std(f,0,1).^2)))
```

```
sum(sum((std(s3,0,1).^2)))/sum(sum((std(f,0,1).^2)))
```

命令行窗口

不熟悉 MATLAB?请参阅有关[快速入门](#)的资源。

ans =

0.9980

ans =

0.9824

ans =

0.9372

#### 四.程序清单

CAM.m

原始图片.bmp

s1.bmp

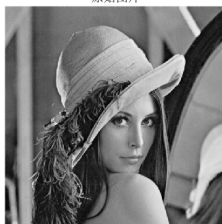
s2.bmp

s3.bmp

untitled.bmp

## 五.运行截图

原始图片



压缩为0.5，信息量为0.9980



压缩为0.125，信息量为0.9824



压缩为0.015625，信息量为0.9372









## 六.参考文献

- 1.PCA降维算法总结以及matlab实现PCA(个人的一点理解) - Work Hard, Play Harder! - CSDN博客——<https://blog.csdn.net/watkinsong/article/details/8234766>
- 2.PCA图像压缩的matlab实现 - izcr的博客 - CSDN博客——<https://blog.csdn.net/xfijun/article/details/51052049>
- 3.PCA与图像压缩 - FireMicrocosm的专栏 - CSDN博客——<https://blog.csdn.net/FireMicrocosm/article/details/47394909>
- 4.Matlab 之 im2col 【转】 - Providence - 博客园——<https://www.cnblogs.com/rong86/p/3557193.html>
- 5.X=[ones(16,1) x]在matlab 中表示什么意思? \_百度知道——<https://zhidao.baidu.com/question/343443293.html>
- 6.matlab 中diag函数的用法 - carrie8899的专栏 - CSDN博客——<https://blog.csdn.net/carrie8899/article/details/8490253>
- 7.MATLAB sort函数用法 - Mieet - 博客园——<https://www.cnblogs.com/zhangziyan/p/8822231.html>
- 8.PCA · 斯坦福机器学习笔记——<https://yoyoyohamapi.gitbooks.io/mit-ml/content/%E7%89%B9%E5%BE%81%E9%99%8D%E7%BB%B4/articles/PCA.html>
- 9.特征值和特征向量 - MATLAB eig - MathWorks 中国——<https://ww2.mathworks.cn/help/matlab/ref/eig.html>
- 10.PCA 原理及其在图像压缩中的应用 - pdpdpd - CSDN博客——[https://blog.csdn.net/sjtu\\_edu\\_cn/article/details/49095917](https://blog.csdn.net/sjtu_edu_cn/article/details/49095917)

11.PCA 降维算法详解 以及代码示例 - Work Hard, Play Harder! - CSDN博客——<https://blog.csdn.net/watkinson/article/details/38536463>

12.matlab中的两种保存图像的方法： saveas imwrite. - brandyzhaowei的专栏 - CSDN博客——<https://blog.csdn.net/brandyzhaowei/article/details/8004659>

13.一分钟了解“用matlab计算图像的熵 entropy” - 一点点滴滴 - CSDN博客——<https://blog.csdn.net/yes1989yes/article/details/81390068>