基于 PCA 图像压缩实验报告

智能科学与技术 16337174 吕高雅馨

➢ 实验内容

使用 PCA 进行图像压缩——输入一张灰度图片 Lena，图片大小为 512*512，使用 PCA 方法把原始图片按照 2：1、8：1、32：1 的压缩率进行压缩，分析压缩后的数据所含信息量大小，并比较压缩数据再已经过重建后与原始图片的视觉差异。

➢ 实验原理

主成分分析(Principal Component Analysis, PCA) 是研究如何通过原来变量的少数几个线性组合来解析原来变量绝大多数信息的一种多元统计方法

通过数据集的协方差矩阵及其特征值分析，我们就可以得到主成分的值。

一旦得到了协方差矩阵的特征值和特征向量，我们就可以保留最大的 N 个特征。这些特征向量也给出了 N 个最重要特征的真实结构，我们就可以通过将数据乘上这 N 个特征向量从而将它转换到新的空间上。

➢ 实验步骤

把图像分割成 1024 个 16*16 的小块，把每一个小块的数据看成一个样本，作为列向量X_i存入一个数据矩阵中；

对这个数据矩阵进行标准化操作，并求协方差，得到相关矩阵；

从相关矩阵出发求解主成分

令 $\Sigma^{1/2}=\begin{pmatrix}\sqrt{\sigma_{11}},0...0\\ 0,\sqrt{\sigma_{22}},...0\\ ...\\ 0,...0,\sqrt{\sigma_{pp}}\end{pmatrix}$　　对原始变量X进行如下标准化

$$Z=(\Sigma^{1/2})^{-1}(X-\mu)$$

$$E(Z)=0 \qquad \mathrm{cov}(Z,Z)=(\Sigma^{1/2})^{-1}\Sigma(\Sigma^{1/2})^{-1}=\begin{pmatrix}1,\rho_{12},...,\rho_{1p}\\ \rho_{21},1,...,\rho_{2p}\\ ...\\ \rho_{p1},\rho_{p2},...,1\end{pmatrix}=R$$

原始变量$X_1,X_2,...X_p$的相关阵实际上就是对原始变量标准化后的协方差矩阵

求特征值和特征向量，并排序；

按照压缩率取前 k 个特征向量（主成分），并计算它们累计的方差贡献率，即压缩数据所包含的信息量百分比；

重构图像，把数据矩阵中的列向量再转回小图像块生成图像。

➢ 实现方法

使用 matlab 编写一个 myPCA 函数，用 PCA 方法实现不同压缩率的图像压缩。

```
function[pcaData, alpha] = myPCA(data, size_block, featuresToExtract);
%输入: data,原始图像的数据
%       size_block，将原始图像切分为正方形小块，小块的边长
%       featuresToExtract,提取的主成分数
%输出: pcaData,压缩后图像的数据
%       alpha,提取的主成分累计方差贡献率，用来表示压缩后的信息量百分比
```

按照前面所述的实验步骤，先将原始图像数据切分成 1024 个 16*16 的小图像块，每一块的维数是 256，因此分配内存给一个 256*1024 的矩阵，将数据存进去。

```
%得出原始图像数据的尺寸
[row col] = size(data);
slice_size = size_block * size_block;       %切割成小图像块的维数
m = 0;
matrix = zeros(slice_size, (row*col)/slice_size);       %256*1024矩阵
for i = 1:size_block:row
    for j = 1:size_block:col
        m = m+1;
        block = data(i:i+size_block-1, j:j+size_block-1);
        matrix(:,m) = block(:);
    end
end
```

计算相关矩阵

```
%标准化处理
normalizedData = matrix - ones(size(matrix,1),1)*mean(matrix);

%求相关矩阵（标准化后的协方差矩阵）
covarianceMatrix = cov(normalizedData');
```

计算特征值和特征向量并排序

```
%求特征值和特征向量
[eigVec, eigVal] = eig(covarianceMatrix);
eigVal = diag(eigVal);
[eigVal t] = sort(eigVal, 'descend');
eigVec = eigVec(:,t);
```

按照压缩率取前 k 个特征向量（主成分）——如果压缩率为 2：1，则 k 的值为 128，如果压缩率为 8：1，则 k 的值为 32，如果压缩率为 32：1，则 k 的值为 8 。并计算它们累计的方差贡献率，即压缩数据所包含的信息量百分比。

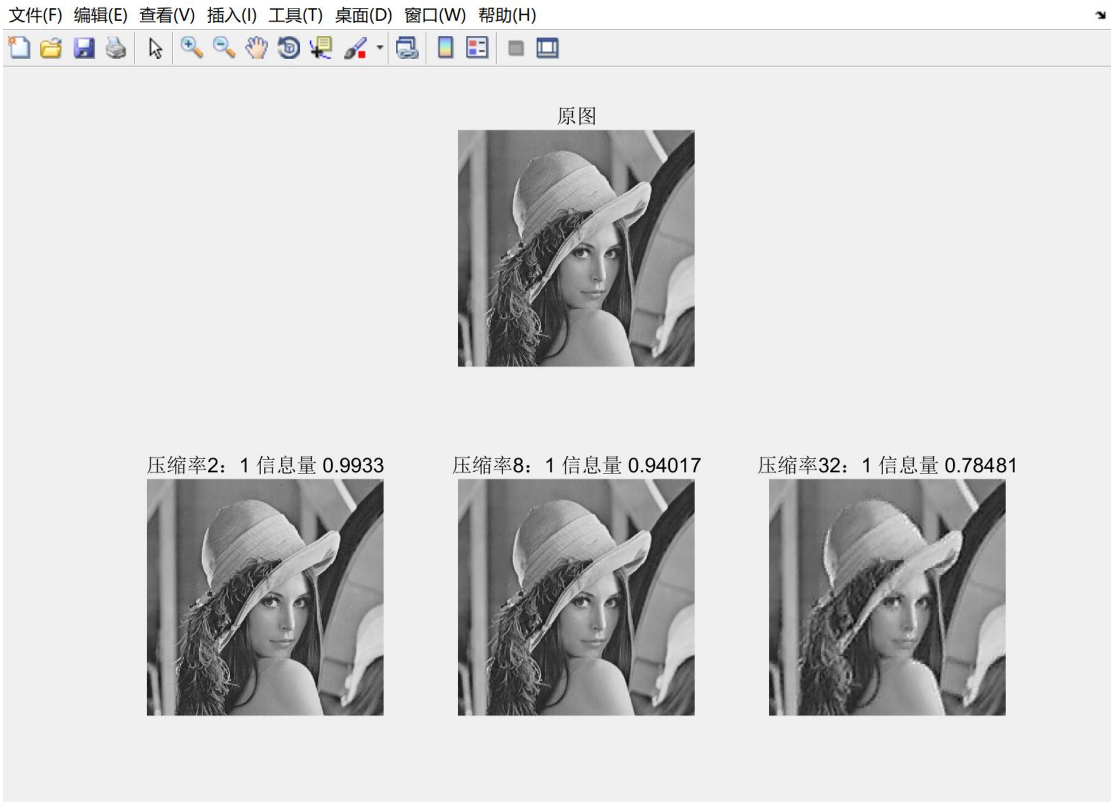
```
projectionVectors = eigVec(:,1:featuresToExtract);
%计算所取特征值贡献率(信息量大小)
alpha = sum(eigVal(1:featuresToExtract))/sum(eigVal);
```

重构图像

```
%重构图像
y = projectionVectors'* matrix;
pcaData = projectionVectors * y + ones(size(projectionVectors,1),1)*mean(matrix);
m= 0;
for i = 1:size_block:row
    for j = 1:size_block:col
        m = m + 1;
        block1 = reshape(pcaData(:,m), size_block, size_block);        % 列向量块转化为方块
        Out(i:i+size_block-1, j:j+size_block-1) = block1;
    end
end
pcaData = Out;
```

➢ 实验结果

运行 compression.m 文件，得到结果如下：



很明显，压缩率约大，选取的主成分越少，重建的图像与原图像相比也就更模糊，通过定量分析压缩后数据所包含的信息量也可以得出结论。