第三次作业

题目：使用 PCA 进行图像压缩

描述：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。

例子：

提示：例子：

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。

压缩率计算：16*16=256 维度，压缩率为原始的 1/2，即变成 128 维度。图片重新生成，是利用 128 维度重构图片块 16*16，重构如上。

举例 matlab 读图：imread, imwrite。

PCA 方法的主要步骤如下：

将所有的样本数据x^i（列向量）拼成一个矩阵 $\{x^1, x^2, \ldots, x^i, \ldots, x^K\}$。

第一步是预处理，要保证数据的均值为 0。那么

$$\mu := \frac{1}{k}\sum_{i=1}^{k} x^i$$

$$x^i := x^i - \mu$$

求这个矩阵的协方差矩阵：

$$\Sigma = \frac{1}{K}\sum_{i=1}^{K}(x^i)(x^i)^T$$

然后求出矩阵 Σ 对应的特征向量和特征值。将特征值按从大到小排列 $\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$，其对应的特征向量为$\{u_1, u_2, \ldots, u_K\}$

可以证明，u_1就是所有数据点的主要分布方向。那么，我们就可以据此排除一些次要的分布方向，保留更重要的分布方向。我们保留前 k 个特征向量，那么它们对应的矩阵

$$\widehat{U} = \{u_1, u_2, \ldots, u_K\}$$

如果选择这 k 个向量作为新的坐标系，那么数据点x^i的坐标是：

$$\{u_1 x^i, u_2 x^i, \ldots, u_k x^i\}^T$$

对数据进行降维：

$$\widehat{x^i} = \widehat{U}\widehat{U}^T x^i$$

那么$\widehat{x^i}$就是将原来的x^i投影在前 k 个最重要的特征向量方向上之后，在原始坐标系中的坐标。

在图像中，选取 L 列进行 PCA 降维，假设原始数据为 N 维，降维到 M 维。在使用训练数据之前，需要对每个数据进行零均值处理。

$$x^i := x^i - \mu^i$$

$$\mu^i := \frac{1}{N}\sum_{j=1}^{N} x^i(j)$$

然后，将降维方法应用到图像的所有列。那么，整张图像就降为 M 维，实现了数据压缩。

matlab 编程如下：

pca_test 函数，压缩和恢复图像

编辑器 - C:\Users\pangyzh\Desktop\pca_test.m

pca_test.m | fun_pca.m

```matlab
function pca_test(P)                        %P为保留的维数
img=imread('Lena.bmp');
figure(1),subplot(121),imshow(img,[]);title('Original Image');
[M, N] = size(img);
f = double(img);
bs = 16;                                    %图像块尺寸
% PCA图像压缩
g = im2col(f, [bs bs], 'distinct'); %将图像块转换成列矢量表示
g_m = ones(size(g,1),1)*mean(g);            %计算每个块的灰度均值
g = g - g_m;                                %进行零均值化
[E, D]=fun_pca(g);                          % E为特征矢量（第一列对应最大特征值）
                                            % D为特征值（按下降顺序排列）
E_proj = E(:,1:P);                          %取最大的p个特征值所对应的特征矢量进行降维
g_proj = g'*E_proj;                         %从bs*bs维映射到p维

% 恢复图像
g_rec = g_proj*E_proj';
s = g_rec' + g_m;
s = col2im(s, [bs bs], [M N], 'distinct');
figure(1),subplot(122),imshow(s,[]);title('Recovered Image');
```

fun_pca 函数，求矩阵特征向量和特征值

编辑器 - C:\Users\pangyzh\Desktop\fun_pca.m

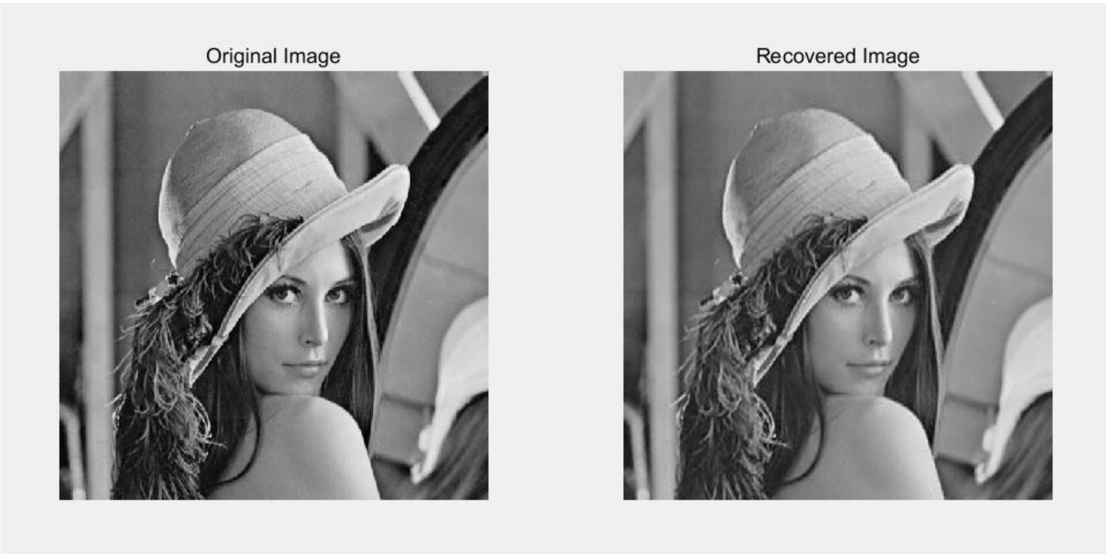
pca_test.m | fun_pca.m

```matlab
function [E,D] = fun_pca(X)
% do PCA on image patches
% INPUT variables:
% X                 matrix with image patches as columns
% OUTPUT variables:
% E                 特征矢量（第一列对应最大特征值）
% D                 特征值（按下降顺序排列）
% Calculate the eigenvalues and eigenvectors
covarianceMatrix = X*X'/(size(X,2)-1);
[E, D] = eig(covarianceMatrix);
% Sort the eigenvalues and recompute matrices
[d_out,order] = sort(diag(D),'descend');
E = E(:,order);
d = diag(D);
D = diag(d(order));
```

结果如下：

1/2 压缩率对比图

1/8 压缩率对比图



1/32 压缩率对比图

心得：PCA 在图像压缩处理中有着重要的应用。假定一副图像 n×n 个像素，如果将这 n×n 个数据一起传送，往往会显得数据量太大。因此，我们希望能够改为传送另外一些比较少的数据，并且在接收端还能够利用这些传送的数据重构原图像。若维数偏小，即压缩比 ρ 偏大，则重构的图像的质量有可能不能令人满意。反之，过大的维数又会导致压缩比过小，从而降低图像压缩和传送的效率。因此，需要根据不同种类的图像，选择合适的压缩比，以兼顾图像传送效率和重构质量。