

基于降维的图像压缩

- 姓名：丛可迪
- 学号：PB21051170

实验原理

紧奇异值分解的定义：

$$\begin{aligned}A_{m \times n} &= U \Sigma V^T \\U U^T &= I_m \\V V^T &= I_n \\\Sigma &= \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \\\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0 \\p &= \min(m, n)\end{aligned}$$

紧奇异值分解是在弗罗贝尼乌斯范数意义下的无损压缩，想做到有损压缩需要采用截断奇异值分解：

$$A_{m \times n} \approx U_k \Sigma_k V_k^T, 0 < k < \text{Rank}(A)$$

矩阵的奇异值会按照从大到小的顺序排列。通过保留较大的奇异值并舍弃较小的奇异值，可以近似表示原矩阵，从而实现数据压缩。

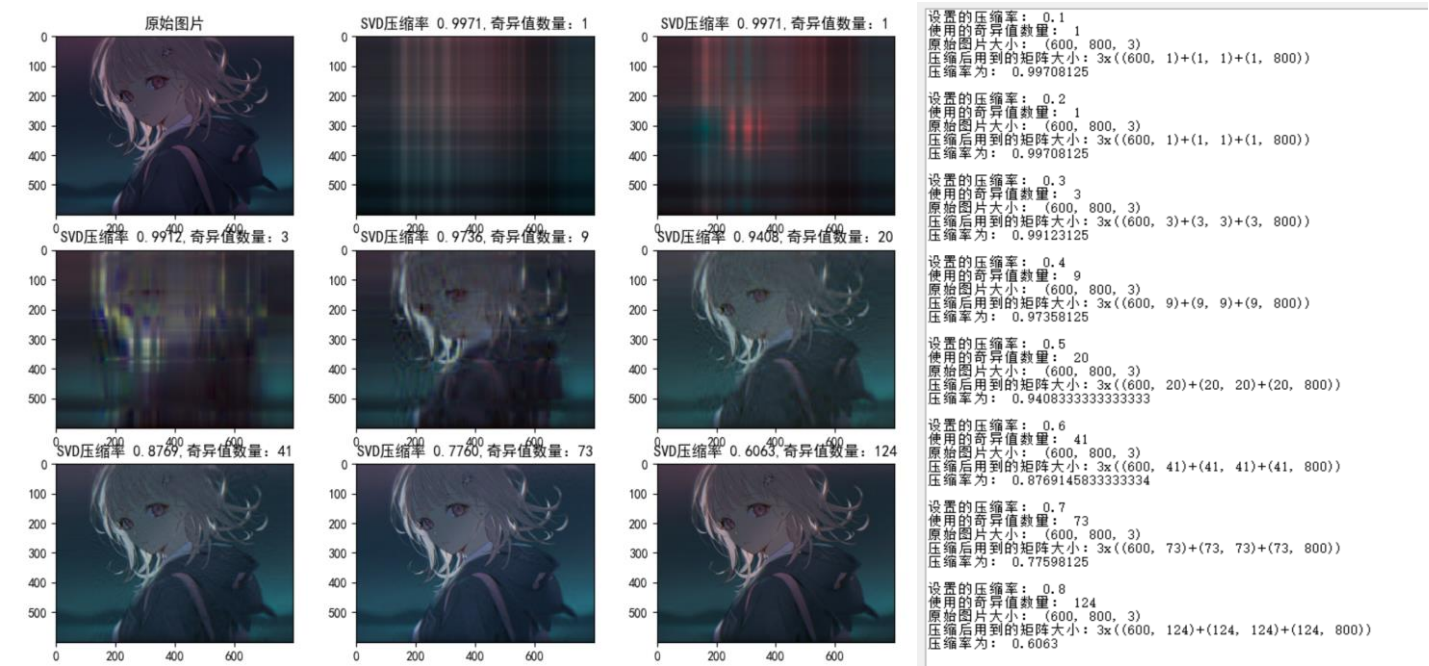
灰度图片有 1 个通道，彩色图片有 3 个通道，也就是矩阵的一个位置上存储了 1/3 个数值。通过对 1/3 个图层矩阵进行 SVD 近似，取前 k 个最大的奇异值进行近似表达，即可用较少的数据去表达图片。

实验代码

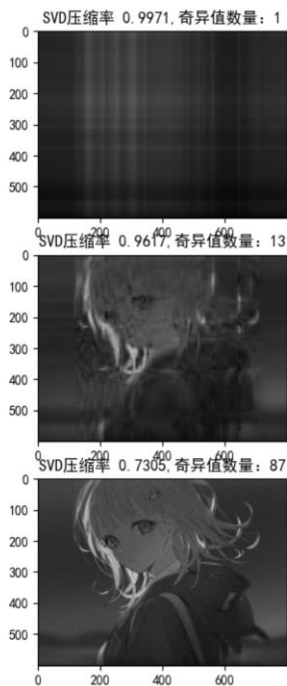
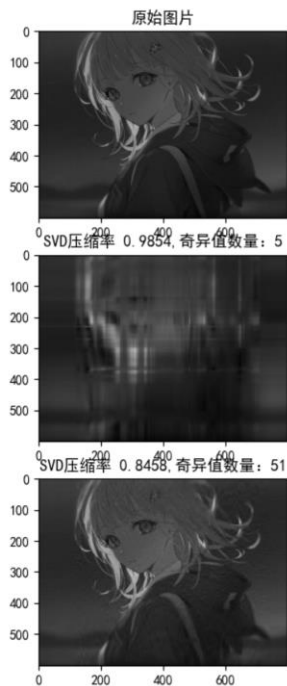
见附件代码文件

实验结果

可执行程序见附件
对彩色图像进行压缩



对灰度图像进行压缩



设置的压缩率: 0.1
使用的奇异值数量: 1
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 1) + (1, 1) + (1, 800))$
压缩率为: 0.99708125

设置的压缩率: 0.2
使用的奇异值数量: 2
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 2) + (2, 2) + (2, 800))$
压缩率为: 0.9941583333333334

设置的压缩率: 0.3
使用的奇异值数量: 5
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 5) + (5, 5) + (5, 800))$
压缩率为: 0.9853645833333333

设置的压缩率: 0.4
使用的奇异值数量: 13
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 13) + (13, 13) + (13, 800))$
压缩率为: 0.96173125

设置的压缩率: 0.5
使用的奇异值数量: 28
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 28) + (28, 28) + (28, 800))$
压缩率为: 0.9167

设置的压缩率: 0.6
使用的奇异值数量: 51
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 51) + (51, 51) + (51, 800))$
压缩率为: 0.84583125

设置的压缩率: 0.7
使用的奇异值数量: 87
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 87) + (87, 87) + (87, 800))$
压缩率为: 0.73048125

设置的压缩率: 0.8
使用的奇异值数量: 140
原始图片大小: (600, 800)
压缩后用到的矩阵大小: $1 \times ((600, 140) + (140, 140) + (140, 800))$
压缩率为: 0.5508333333333333