

# SVD 在图像压缩上的应用

作者: 谢唯嘉 学号: 6120210299

专业班级: 信息与通信工程信研 1 班

---

## 摘要

在很多情况下, 数据的一小部分包含了数据的绝大部分信息, 用线性代数的许多矩阵分解方法都可以将矩阵特征表现成新的易于理解的形式, 在这一过程之中, 实际是一个提取数据集特征值的过程 (降噪), 有利于我们提高算法的精准度, 减少数据运算量。最常见的矩阵分解算法就是矩阵奇异值分解 (SVD), SVD 在图像压缩、推荐系统、金融数学等领域都有应用, 著名的主成分分析算法 (PCA) 也是通过 SVD 实现的。

关键词: SVD 压缩 图像

---

## 一、引言

数字图像的数据量庞大, 存在较强的内在相关性, 所以有必要对数字图像进行压缩 (编码), 减少其相关性, 形成高效的表示方法。图像压缩对于人们在存储空间较小的计算机中存储图像和在某个网页中更快地加载图像来说是非常必要的 [1]。随着信息通信技术和互联网发展的新发展, 包括数字化和无损压缩技术在内的这一问题在不久的将来可以得到完全解决 [2]。事实上, 我们在互联网上遇到的大多数图片都是经过压缩的 [3]。例如, jpeg、png 和 gif 格式都是压缩的。虽然图像压缩不可避免地会导致图像质量的下降, 但是一些琐碎的字符却不值得人们关注。因此, 相比图像质量的一点点损失和图像存储空间的急剧减少, 我们更喜欢压缩的图像。在对这个问题进行了一些研究之后, 我们打算使用奇异值分解 (SVD) 的知识。通过选取一些较大的奇异值和相应的左右奇异向量, 我们可以用较少的数据来表示退化较小的图像。

## 二、图像压缩的理论基础

### 2.1 图像存储原理

图像其实是利用二维坐标指定的空间数据。摄像机所获得的亮度转换成信号, 在经过 A/D 转换器处理, 作为一个值存储在计算机内, 可通过图像坐标  $x, y$  进行查询。由此可见计算机以矩阵的形式存储图像, 而图像为一个点阵。对于 RGB 图像, 每幅图像都匹配一个三维张量, 这个张量可以看作是三个矩阵的组合。这个张量中三个矩阵的大小与原始图片大小一样。在计算机中, 有三个从 0 到 255 的整数对应于图像的每个像素。这三个整数的值分别表示像素在 R-通道, G-通道和 B-通道中的属性。实际上, 整数首先

被规范化为从 0 到 1 的浮点数，每个浮点数占用计算机存储的 1 字节。对于一个 RGB 类型的图片有  $m \times n$  个像素点，我们需要大约  $3 \times m \times n$  比特进行存储。

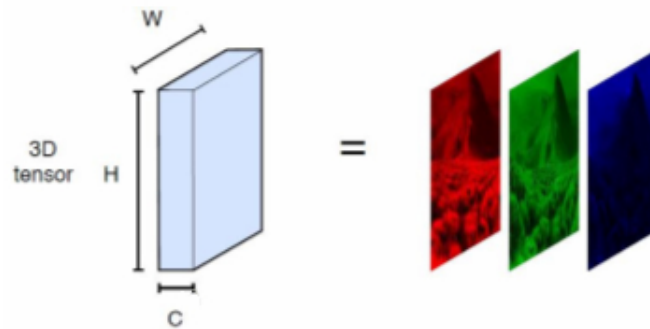


图 1 图片中像素可分为 **RGB** 三通道

## 2.2 奇异值分解的原理

### 2.2.1 特征值和对角化

#### 特征值和特征向量

如果一个  $v$  是方阵  $A$  的特征向量，我们可以得到以下形式：

$$Av = \lambda v$$

$\lambda$  被称为特征值， $v$  被称为特征向量。

#### 对角化

如果  $n \times n$  方阵  $A$  有  $n$  个线性无关的特征向量，则矩阵  $A$  可以对角化，这意味着我们可以得到以下形式：

$$P^{-1}AP = \Lambda$$

$P$  的列是  $A$  的特征向量，矩阵  $\Lambda$  是一个对角矩阵，对角项分别是  $A$  对应于  $P$  中的特征向量的特征值。

### 2.2.2 对称矩阵与正交可对角化矩阵

如果  $A$  是对称矩阵 (即  $A=A^T$ )，它可以被正交对角化。

$$Q^{-1}AQ = \Lambda$$

其中  $Q$  是正交矩阵 ( $Q^{-1} = Q^T$ )。

或者另一种形式：

$$A = Q\Sigma Q^{-1}$$

这是一种特殊的特征值分解。

### 2.2.3 奇异值分解 (SVD)

之前介绍的特征值分解只能应用于某些方阵，这意味着它是有限的。因此，如果我们想要得到一个更一般的矩阵的分解方式 (例如，一个  $m \times n$  矩阵， $m \neq n$ ) 我们应该找另一种方法 [4]。

我们假设矩阵  $A$  可以分解为以下形式：

$$A = U_{m \times m} \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_\gamma & \\ & & & 0 \end{pmatrix}_{m \times n} V_{n \times n}^T = U \Sigma V^T$$

( $U$  和  $V$  是标准正交矩阵)

SVD 分解的目的，如果有接近 0 的奇异值，可以不进行计算，一方面可以减少计算量，另一方面可以过滤噪声。

### 2.3 SVD 在图像压缩上的应用

我们知道任何 RGB 模型图像都可以存储到一个具有 3 个通道的 3D 张量中。因此，我们可以压缩每个通道中的矩阵，并将它们组合在一起得到压缩图像。如前所述，任何  $m \times n$  矩阵都可以分解为以下形式：

$$A_{m \times n} = U_{m \times m} \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_\gamma & \\ & & & 0 \end{pmatrix}_{m \times n} V_{n \times n}^T = U \Sigma V^T$$

我们可以用谱分解的形式重写这个矩阵。奇异值可以看作是不同矩阵的权值。而奇异值越大的分解子矩阵对结果的影响越大，提供的信息也越多。因此，在进行奇异值分解时，我们倾向于将较大的奇异值重新排列在前面。因此，我们只需要选择前  $k$  个奇异值及其对应的向量，就可以代表退化程度相对较小的原始图像。假设我们有一个  $m \times n$  像素的图像。然后我们需要  $3 \times m \times n$  比特去存储它。然而，在进行奇异值分解并选择前  $k$  个奇异值后，它只会占用  $3k(m+n+1)$  字节。事实上， $k \ll m$  和  $k \ll n$ ，因此图像的存储空间大大缩小，较为直观的表现在于图像的大小成倍减少。我们设  $\frac{k(m+n+1)}{m \times n}$  为压缩比。

## 三、图像压缩实验

### 3.1 代码实现

整体 SVD 在图像压缩上的应用代码基于 Python 语言进行编程。通过 numpy 函数库进行矩阵的各种运算操作，通过 pillow 函数库对于处理好的图像进行显现与保存。具体运行代码如下：

#### 图像压缩

```
1 import numpy as np
2 from PIL import Image
3
4 # 使用奇异值总和的百分比进行筛选
5 def svd(data,scale):
6     # scale 代表你要保留的奇异值比例
7     u,sigma,v = np.linalg.svd(data)
8     svd_data = np.zeros(data.shape)
9     total = sum(sigma)
10    sum_data = 0
11    for index,item in enumerate(sigma):
12        svd_data += item * np.dot(u[:,index].reshape(-1,1),v[index,:].reshape(1,-1))
13        sum_data += item
14        if sum_data >= scale * total:
15            break
16    return svd_data
17
18 def compress(data,scale):
19     #获得图片RGB三通道数并进行压缩
20     r = svd(data[:, :,0],scale)
21     g = svd(data[:, :,1],scale)
22     b = svd(data[:, :,2],scale)
23
24     result = np.stack((r,g,b),2)
25     result[result > 255] = 255
26     result[result < 0] = 0
27     result = result.astype(int)
28     return result
29
30 if __name__ == '__main__':
31     image = Image.open('supercar.jpeg')
32     width,height = image.size
33     arr = np.zeros((width,height,3)) # RGB
34     for x in range(width):
35         for y in range(height):
36             arr[x][y] = image.getpixel((x, y))
37     # 原生 range 不支持浮点数，所以用 np.arange 代替
38     for c in np.arange(.1,1.,.2): #(.1,.9,.2)
39         result = compress(arr,c)
40         for x in range(width):
41             for y in range(height):
42                 image.putpixel((x, y),(result[x,y,0],result[x,y,1],result[x,y,2]))
43     image.save('test_'+str(int(100 * c))+'.jpeg')
```

上述代码具体操作步骤:

**第一步:** 利用 PIL 中的 Image 模块将要进行压缩的图片进行读取, 并获得图片的宽和高。

**第二步:** 利用 PIL 中的 getpixel 模块获取输入图像中的某一点的像素的 RGB 颜色值。

**第三步:** 程序进入图片压缩循环, 通过设置要保存的奇异值比例对图像进行压缩处理并进行保存图片。

## 3.2 实验结果与评价指标

### 3.2.1 实验结果

整个实验中用图来自网络用图, 调用代码生成拥有不同保留比例奇异值的图片。以下为实验结果生成的图2。



图 2 原始图像与不同特征值保留比例压缩图像

### 3.2.2 主观评价方法

人们对一幅图片视觉感受是主观评价, 人自身对图像的评价是最为准确的。对于这种主观评价方法能参考国际电联 ITU-R 关于电视图像主观质量评价 BT.500-13 标准中的平均意见得分 (MOS) 方法 (如图3), 就是通过人来观察图像, 对图像的优劣进行主观评定 (打分)。图像质量主观评价的 MOS 计分有两个尺度, 即国际上通行的 5 级评分的质量尺度和损伤尺度。

Mean opinion score (MOS)		
MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

图 3 图像主观质量的 MOS 分评价

作者主观通过图2原始图片与压缩完后的图片进行比较，不难发现对于原始图片压缩后只保留 10%,30% 和 50% 的特征值直观上对于图片所要表达的信息有所缺失。相反的如果保留 70% 和 90% 的特征值的图片，依旧能够很好的表达图片的内容。主观的根据 MOS 分评价体系 (图3) 对上述不同特征值保留的图片进行打分如下表1：

图片	保留 10%	保留 30%	保留 50%	保留 70%	保留 90%
得分	1	2	3	4	5

表 1 图片主观打分

### 3.2.3 客观评价方法

由于本实验的实验目的是对于照片的压缩储存处理，故引入照片存储大小和尺寸两个对比因素。具体对比结果见下表2。

图片	原图	保留 90%	保留 70%	保留 50%	保留 30%	保留 10%
尺寸	2000x1125	2000x1125	2000x1125	2000x1125	2000x1125	2000x1125
大小	1.4 MB	261.2 KB	230.7 KB	163.9 KB	91.9 KB	78.4 KB

表 2 各基于特征值的压缩比例图片对比

本实验的压缩操作对于图片的尺寸没有进行压缩操作，而对于大小有较为明显的压缩。压缩完后图片大小与原始图片大小比例如下表3。

图片	原图	保留 90%	保留 70%	保留 50%	保留 30%	保留 10%
大小	1.4 MB	261.2 KB	230.7 KB	163.9 KB	91.9 KB	78.4 KB
大小压缩比例	100%	18.2%	16.1%	11.4%	6.4%	5.5%

表 3 各基于特征值的压缩比例大小压缩百分比

#### 四、结论

综上所述，对于需要储存的大数据图片，SVD 压缩是降低图片数据量的有效方法，而且依旧能够识别出重要的图像细节。它可以像滤波器一样用于图像去噪 [5]。通过维纳滤波处理，SVD 压缩速度更快。因此，与全秩图像相比，压缩图像在数据库中需要更少的计算机存储和传输时间，具有很大的应用场景。

## 参考文献

- [1] A. Ranade, S. S. Mahabalarao, and S. Kale, “A variation on svd based image compression,” *Image and Vision computing*, vol. 25, no. 6, pp. 771–777, 2007.
- [2] M. Nixon and A. Aguado, *Feature extraction and image processing for computer vision*. Academic press, 2019.
- [3] 朱秀昌, 唐贵进, 现代数字图像处理. 人民邮电出版社, 2020.
- [4] 杨明, 刘先忠, 矩阵论 (第 2 版). 华中科技大学出版社, 2005.
- [5] V. Kolev, “Big data in astroinformatics—compression of scanned astronomical photographic plates,” *arXiv preprint arXiv:2108.08399*, 2021.



## 五、对老师评价

时光飞逝，不知不觉和邹老师度过了两个月的学习时间，在和老师学习的过程中受益匪浅。我能从老师身上学到许多课程之外的知识，教学方式有特色，教学具有启发式，为后续课程学习与实践训练打下很好的基础。老师幽默风趣，上课气氛活跃。老师和学生的互动性得到了充分的体现。我们从他那里学到的不仅仅是科学文化知识，更有学习方法和做人的道理。至于不好的方面，基本上没有，如果要挑的话，那就是在 PPT 制作过程中存在少数错误，偶尔会在课堂上扯到国际形势。