



# 数学建模 图像颜色变换 彩色图像灰度化

姓名 徐海阳 学号 PB20000326 院系 少年班学院

2023 年 3 月 17 日

## 摘 要

当我们处理彩色图像时，有时候需要将彩色图像转化为灰度图像。灰度图像是一种只包含亮度信息而不包含颜色信息的图像。彩色图像中的每个像素通常由 RGB 三个通道的颜色值组成，而灰度图像中每个像素只有一个亮度值。将彩色图像转换为灰度图像可以减少图像处理的复杂度和计算量，同时也能够更好地突出图像中的亮度变化和纹理信息。本文将介绍三类常见的彩色图像灰度化方法，包括加权平均法、PCA 主成分分析法以及论文《Color Image to Grayscale Image Conversion》[1] 中的方法。通过比较不同方法的优缺点，我们可以选择适合我们应用场景的灰度化方法。

## 目录

1 前言	2
2 相关工作	2
3 数学模型建立	2
3.1 加权平均法	2
3.2 Color Image to Grayscale Image Conversion	3
3.3 PCA 主成分分析	4
4 结果	4
5 结论	4
6 问题	4

## 1 前言

在数字图像处理领域，彩色图像的处理是一个重要的研究方向。然而，在一些应用场景中，需要将彩色图像转换为灰度图像，例如图像处理、计算机视觉、机器学习等领域。这是因为灰度图像具有单通道、处理简单、占用空间小等优点，更适合于某些特定的应用场景。因此，彩色图像灰度化的研究具有重要的实际意义和应用价值。传统的灰度化方法通常基于对每个像素点的 RGB 颜色分量进行某种计算，得到其对应的灰度值。然而，不同的灰度化方法具有不同的计算公式和处理策略，因此会对图像的质量、效率、应用场景等产生不同的影响。因此，研究彩色图像灰度化方法，优化算法并提高图像处理效果，是数字图像处理领域中的研究热点之一。

## 2 相关工作

最简单的灰度化方法是平均值法，它是将每个像素点的 RGB 颜色分量的平均值作为其灰度值。这种方法实现简单，但会忽略颜色通道之间的相互影响；为了考虑到颜色通道之间的相互影响，加权平均值法被提出。该方法不仅将三个颜色通道的值加权平均，而且权值的选择也非常重要。不同的权值会产生不同的灰度图像，因此需要根据实际情况来选择权值。

此外，颜色空间的转换也是一种有效方法。基于颜色空间的灰度化方法是一种将 RGB 颜色空间转换为其他颜色空间的方法。例如，将 RGB 颜色空间转换为 YCbCr 颜色空间，并将 Y 通道作为灰度图像。同时，统计方法也可以帮助我们进行灰度化操作。如，最大值法和最小值法是两种基于灰度值的统计方法。最大值法将 RGB 颜色分量中的最大值作为灰度值，而最小值法则取最小值。这两种方法很简单，然而却不能很好地反映图像中的亮度和纹理信息。

## 3 数学模型建立

GRB (Green-Red-Blue) 是一种基于三原色 (红、绿、蓝) 的颜色模型，广泛应用于计算机图形学和数字图像处理中。在彩色图像中，每个像素点由三个通道 (R、G、B) 的数值表示，这三个通道的数值分别代表了像素的红色、绿色和蓝色成分的强度，通常取值范围为 0 到 255，其中 0 表示最小强度，255 表示最大强度。因此，一幅二维图片可以由一个形状为  $[H, W, 3]$  的张量  $\mathbf{X}$  来表示，其中  $H$  代表图片的高， $W$  代表图片的宽，3 即为 RGB 这 3 个通道。具体来说， $\mathbf{X}[h, w, c]$  代表图片中  $[h, w]$  处像素在  $c$  通道上的取值，如  $\mathbf{X}[0, 0, 0]$  就代表图片最左上角那个像素在 Red 通道上的取值。

因此，我们的数学模型如下：给定一个张量  $\mathbf{X} \in [H, W, 3]$ ，其中最后一个维度的取值在  $[0, 255]$ 。我们希望通过某种方式，将其转化为  $\mathbf{X} \in [H, W, 1]$ ，最后一个维度的取值也在  $[0, 255]$ 。

### 3.1 加权平均法

简单的平均值法、加权平均值法和 OpenCV 中的函数 `cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)` 都属于这一类方法。这类方法的本质思想是对 R、G、B 三个通道的信息 (数值) 进行融合 (线性组合)，线性系数如表 1 中所示：

平均值法的考量非常简单，直接将 R、G、B 三个通道的值平均一下得到灰度值。

加权平均法考虑到人眼对于红绿蓝三种颜色的感知是有差异的，对于绿色更为敏感。因此，在加权平均的过程中提高了 G 通道的权重。具体来说，RGB 的权重为  $(0.299, 0.587, 0.114)$ 。

`cv2.COLOR_RGB2GRAY` 方法也使用了  $(0.299, 0.587, 0.114)$  权重。

方法	R	G	B
平均值法	0.333	0.333	0.333
加权平均值法	0.299	0.587	0.114
cv2.COLOR_RGB2GRAY	0.299	0.587	0.114

表 1: 加权平均法权值

### 3.2 Color Image to Grayscale Image Conversion

这篇论文中的算法很清晰，在第 3 部分 Algorithm 中，如下算法所示：

---

**Require:** A color image represented by its RGB components  $R, G, B$ .

**Ensure:** A grayscale image represented by its intensity component  $I$ .

- 1: **procedure** COLORTOGRAYSCALE( $R, G, B$ )
- 2:   Calculate luminance value:  $Y \leftarrow (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$
- 3:   Calculate blue chrominance value:  $U \leftarrow (B - Y) \times 0.565$
- 4:   Calculate red chrominance value:  $V \leftarrow (R - Y) \times 0.713$
- 5:   Calculate sum of chrominance values:  $UV \leftarrow U + V$
- 6:   Approximate red component:  $R1 \leftarrow R \times 0.299$
- 7:   Approximate green component:  $G1 \leftarrow G \times 0.299$
- 8:   Approximate blue component:  $B1 \leftarrow B \times 0.299$
- 9:   Approximate red component:  $R2 \leftarrow R \times 0.587$
- 10:   Approximate green component:  $G2 \leftarrow G \times 0.587$
- 11:   Approximate blue component:  $B2 \leftarrow B \times 0.587$
- 12:   Approximate red component:  $R3 \leftarrow R \times 0.114$
- 13:   Approximate green component:  $G3 \leftarrow G \times 0.114$
- 14:   Approximate blue component:  $B3 \leftarrow B \times 0.114$
- 15:   Compute average of the approximated RGB values to obtain grayscale value:

$$R4 = (R1 + R2 + R3)/3$$

$$G4 = (G1 + G2 + G3)/3$$

$$B4 = (B1 + B2 + B3)/3$$

$$I1 = (R4 + G4 + B4 + UV)/4$$

16: **end procedure**

---

这个算法的设计是基于以下考虑：

1. 人眼对亮度的敏感度比对色彩的敏感度更高，因此可以使用亮度值来表示灰度图像。
2. 为了保留原始图像中的颜色信息，需要将原始图像中的色彩信息转换为灰度图像中的亮度和色差信息。本算法使用 YUV 颜色空间来实现这一目标。
3. 为了保持灰度图像与原始彩色图像之间的视觉相似性，需要在计算灰度值时保留原始图像中的对比度、锐度、阴影和结构等特征。本算法通过近似 RGB 分量并计算其平均值来实现这一目标。因此，该算法能够在保留原始彩色图像中重要特征的同时，将其转换为高质量的灰度图像。

### 3.3 PCA 主成分分析

由于建模模型本质上是将数据的最后一个维度由 3 降为 1，因此理所当然地可以想到 PCA 法。

首先，简单介绍一下 PCA 算法。PCA (Principal Component Analysis) 是一种常用的数据降维技术，它可以将高维数据转换为低维数据，同时保留原始数据中的大部分信息。PCA 的基本思想是将原始数据投影到一个新的坐标系中，使得投影后的数据具有最大的方差。这个新坐标系由一组正交基向量组成，称为主成分。

PCA 算法的具体步骤如下：

1. 对原始数据进行中心化处理，即将每个特征值减去该特征值在所有样本上的平均值。
2. 计算协方差矩阵，即将中心化后的数据矩阵乘以其转置矩阵。
3. 对协方差矩阵进行特征值分解，得到一组特征向量和对应的特征值。
4. 将特征向量按照对应的特征值大小排序，选择前  $k$  个特征向量作为新坐标系的基向量。
5. 将原始数据投影到新坐标系中，得到降维后的低维数据。

因此，只要将每个 pixel 视作一个样本，然后在一张图所有的 pixel 上做 PCA 降维，把 RGB 3 维降为 Gray 1 维即可。

## 4 结果

如图1所示。给出了 9 张图片在 5 种方法，分别是平均值法、加权平均法、OpenCV 包（等价于加权平均）、PCA 主成分分析和 [1] 的灰度化结果。

## 5 结论

与加权平均法和 [1] 相比，PCA 方法的效果最好。如在倒数第 3 幅图莫奈的《日出印象》中，PCA 方法比其余方法好在可以明显看出太阳，而其余方法看不太出来。

## 6 问题

加权平均方法与 [1] 不必赘述，PCA 算法尽管效果不错，但存在 2 个问题：

1. 不难发现第 5 幅图的 PCA 结果是原图的 inverse（黑白颠倒）。这不难理解：PCA 包含求特征值的过程，numpy 算特征值默认返回正的特征值。这样的话，特征向量对应的就是正的特征值。这就会导致特征向量在将图片从高维投影至低维的过程中，得到的值可能和我们想要的值一样（我们想要的特征向量对应的特征值不一样是正的），也可能是相反数（如第 5 幅图所示）。因此，为了确保效果，我们可以用 PCA 对原图和负原图都做操作，然后取其中符合人类视觉的一幅即可。

2. PCA 将一幅图中所有 pixel 展平为  $[H*W, 3]$  的形状，然后降维。这等价于利用了图像的 global 信息。从这个角度来看，加权平均的方法等价于每个 pixel 只关注自身（自身的 3 个 channel 进行线性组合得到 gray channel）。然而，我想指出，这两种方法都忽略了一个事实：图像信息既不是像 PCA 那样的 global，也不是像加权平均那样的 per-pixel；而应该是 local 的，比如每个 pixel 和它  $7 \times 7$  邻域范围内的 pixels 的关系比较大（互信息大），而不是和所有 pixels（计算量冗余，很多 distant 的 pixels 与当前 pixel 几乎没有关系）或者仅仅与自身有关（周边的 pixels 和当前 pixel 一起组成的 local perceptual field 是很重要的）。如果用 learning-based method 可以很好的处理这个问题：用卷积神经网络来提取图像特征即可，保证了图像信息的 local 特性。

## 参考文献

- [1] C. Saravanan, “Color image to grayscale image conversion,” in *2010 Second International Conference on Computer Engineering and Applications*, vol. 2, pp. 196–199, 2010.



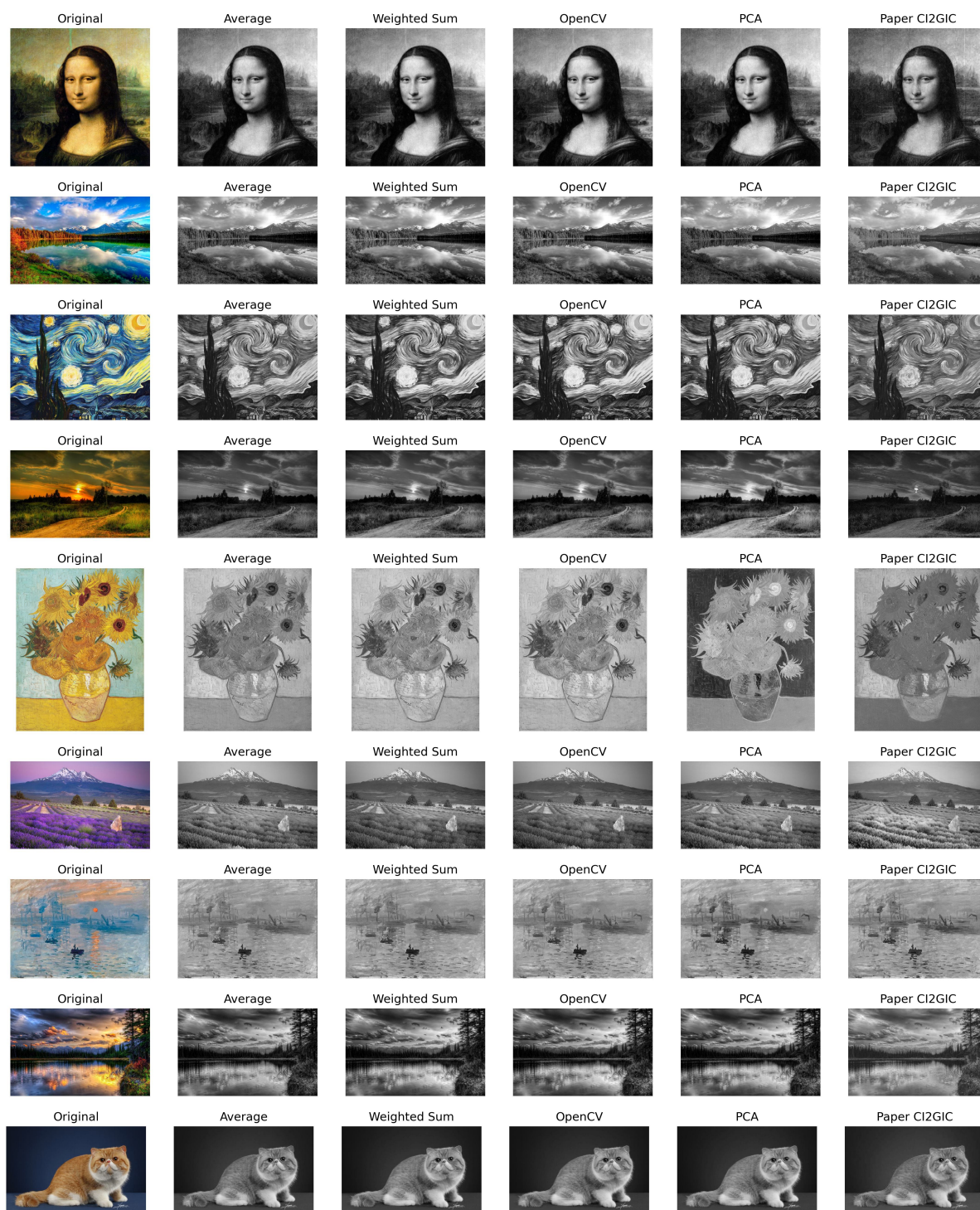


图 1: 原图 (Original) 与 5 种 color2gray 方法的效果展示