

COMP9517: 计算机视觉 2023 T2

实验室 1 规格 最高可得分数:

2.5分

本实验占课程总分值的2.5%。

实验文件应在网上提交。

提交说明将在接近截止日期时公布。

提交的截止日期是第三周，2023年6月14日星期三，18:00:00。

目标：本实验重温了第一周和第二周讲座中涉及的重要概念，旨在使你熟悉实现特定的算法。

材料：本实验室所有问题中所使用的样本图像都可以在WebCMS3中找到。你需要使用OpenCV 3+与Python 3+。

提交：下面的问题4是可以在实验后评估的。在上述截止日期前以Jupyter笔记本（.ipynb）的形式提交本问题的源代码，包括所有的输出（见下面的编码要求）。提交链接将在适当的时候公布。问题1-3是为自己获得图像处理的经验而做的练习，不会被评估，所以你不必提交这些问题的代码。只有问题4将被评估。

1. 对比性拉伸

对比度是衡量图像中强度值的范围，被定义为最大像素值和最小像素值之间的差异。一个8位图像的最大可能对比度是255（最大）-0（最小）=255。任何小于这个值的意味着该图像的对比度比可能的低。对比度拉伸试图通过使用线性缩放拉伸强度值的范围来改善图像的对比度。

假设 I 是原始输入图像， O 是输出图像。让 min 和 max 为允许的最小和最大像素值（对于8位图像，意味着 $min=0$ ， $max=255$ ），让 $imin$ 和 $imax$ 为在 I 中发现的最小和最大像素值。那么经过对比度拉伸的图像 O 就由函数给出：

$$f(x, y) = (f(x, y) \frac{b - a}{d \epsilon}) + \frac{a}{2} - \frac{b}{2} \quad (1)$$

问题1： 写一个算法，按上述公式（1）执行对比度拉伸。

读取给定的图片 **Landscape.png**，并运行你的算法，看看它是否真的改善了对比度。注意，这是一张彩色图像，有三个通道（R、G、B），所以你需要以某种方式将你的算法应用于这三个通道。

这里有不同的可能性。最直接的方法是将其分别应用于每个通道。这是否会产生一个良好的对比度拉伸的彩色图像？

或者，你可以将彩色图像转换为不同的颜色空间，如HSV，并在值（V）通道上计算映射函数（1），然后将其应用于原始图像通道（R、G、B）。这是否会产生一个更好的对比度拉伸的彩色图像？



例子：

输入



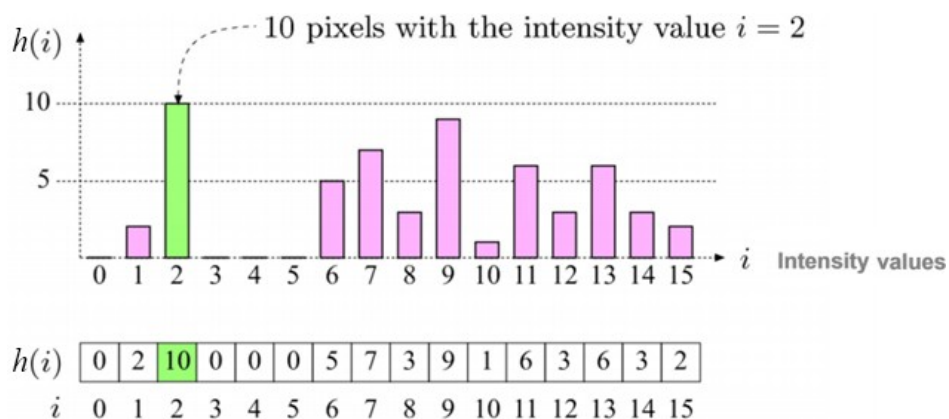
输出

还要写一个算法，找到在任何给定的图像通道中具有最小值的第一个像素（从左上到右下扫描）的坐标和具有最大值的第一个像素的坐标。不要使用现有的库函数来完成这些任务，而要编写你自己的代码（见下面的编码要求）。在输入图像和输出图像上运行你的算法，并打印这两个像素的值，以确认你的对比度拉伸算法是否正确工作。

2. 柱状图的计算

一个图像的直方图显示了强度值的计数。它只给出了像素的统计信息，并删除了位置信息。对于一个具有 \mathbb{N} 的数字图像来说

灰度等级，从0到 $L-1$ ，直方图是一个离散函数 $h(i) = n_i$ 其中 $i \in [0, L - 1]$ 是第 i 个灰度等级， n_i 是该灰度等级的像素数。

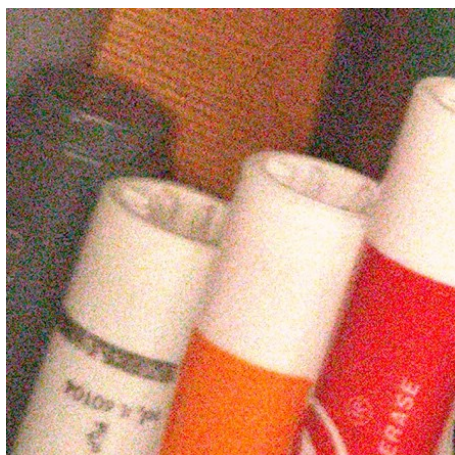


问题2： 写一个算法来计算图像的直方图。不要使用现有的库函数来计算直方图，而是写自己的代码来执行这个任务（见下面的编码要求）。然后在给定的图像**Baboon.png**上运行你的算法，并绘制R、G、B通道的直方图（你可以使用现有的函数来绘制由你的算法计算的直方图阵列）。

3. 图像平滑

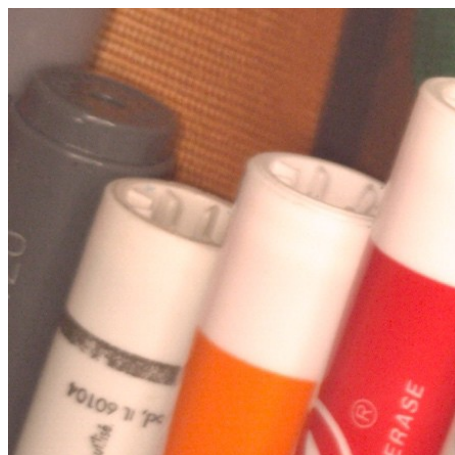
减少图像中的噪声（随机强度变化）可以通过图像过滤来实现。盐和胡椒噪声、脉冲噪声和高斯噪声是图像中一些常见的噪声类型。不同类型的平滑滤波器适用于不同类型的噪声。平滑滤波器通常被称为低通滤波器，因为它允许图像的低频成分通过（具有相似强度值的区域），同时抑制高频成分（边缘或噪声）。

问题3： 实现一个平均（均匀）滤波器和一个中值滤波器。不要使用现有的卷积和图像滤波的库函数，而要自己写代码来完成这个任务（见下面的编码要求）。在给定的图像**Noisy.png**上进行去噪。试着用不同大小的过滤器，观察结果的差异，并决定哪一个是该图像的最佳过滤器和内核大小。



例子:

输入



经过过滤

4. 边缘检测

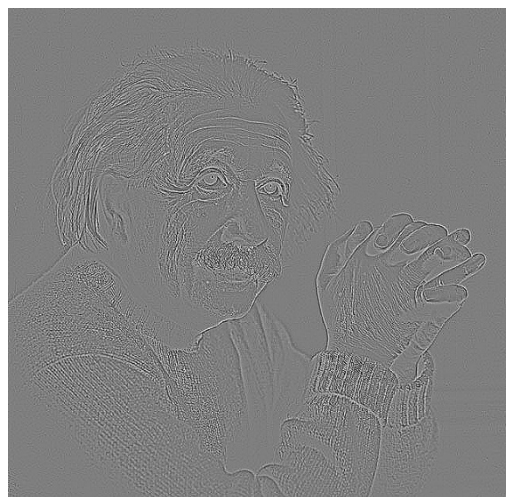
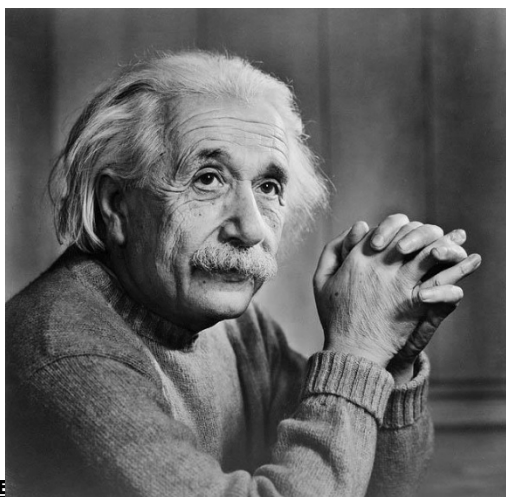
边缘是图像中语义信息的一个重要来源。在人类的视觉感知中，它们出现在不同强度、颜色或纹理的区域之间的分界处。一个灰度图像可以被认为是一个二维景观，在不同的高度有不同强度的区域。拉普拉斯是一个二阶导数算子，可以用来寻找边缘。它强调强度变化大的区域的像素，不强调强度变化慢的区域的像素。一个简单的3×3像素卷积核，近似于拉普拉斯算子，如下：

$$\begin{bmatrix} 0 & 1 & 0 \\ \blacklozenge 1 & -4 & 1 \blacklozenge \\ 0 & 1 & 0 \end{bmatrix}$$

问题4 (2.5分)： 写一个算法，使用上述内核计算输入图像的拉普拉斯图像。使用给定的图像Einstein.png来测试你的算法。不要使用现有的库函数来计算拉普拉斯图像，而是写自己的代码来执行这个任务（见下面的编码要求）。

注意，计算可能产生负的输出像素值。因此，请确保你为计算和输出图像使用正确的数据类型，并使用正确的强度映射来显示输出图像（见下面的例子）。

最后的结果应该是这样的（输出的图像已经过对比度增强）： 输入 输出



编码要求

对于本实验室的所有任务，请自己实现所需的算法，不要使用现有的库函数（来自OpenCV或任何其他软件包）来完成这些任务，除非另有说明。使用这些函数而不是你自己的函数将导致被扣分。

具体来说，你应该自己用普通的Python代码实现以下操作

而不依赖直接执行这些操作（部分）的现有函数：

- 找到最小和最大的像素值（在像素上写自己的循环）。
- 进行对比度拉伸（在像素上写出你自己的循环，以修改它们的值）。
- 计算具有给定值的像素数量（以计算图像的直方图）。
- 用一个小核对图像进行卷积（自己写循环）。
- 对图像应用基本的数学运算（写自己的循环）。对于其他操作，

你可以使用现有的库函数。

在未来的实验室和小组项目中，你可以使用现有的库函数，但在这个实验室中（就像在作业中一样），目标是学习基本的图像处理操作在像素级的工作，并获得自己实现它们的经验。当然，你可以使用现有的函数来验证你自己实现的结果。

确保在你的Jupyter笔记本中，输入图像可以从作为参数指定的位置读取，所有输出图像和其他要求的结果都显示在笔记本环境中。你的笔记本中的所有单元都应该已经执行，这样导师/标记者就不必再次执行笔记本来查看结果。

版权所有：新南威尔士大学CSE COMP9517团队。复制、出版、张贴、分发或翻译本实验作业是对版权的侵犯，并将被提交给新南威尔士大学学生行为与诚信处处理。

发布：2023年6月7日