

# COMP9517: 计算机视觉 2022 T2

## 实验室 1 规格 最高分数: 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。  
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。  
提交说明将在接近截止日期时发布。

提交的截止日期是第3周，2022年6月14日星期二，18:00:00。

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，将此问题的源代码以Jupyter笔记本（.ipynb）和输出图片（.png）的形式提交到一个压缩文件中。提交链接将在适当的时候公布。问题1-3是对自己的练习，以获得图像处理的经验。

### 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位图像的完全对比度是255（最大）-0（最小）=255。任何小于该值的值都意味着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 $II$ 是原始输入图像， $OO$ 是输出图像。让 $a$ 和 $bb$ 为允许的最小和最大像素值（对于8位图像，这意味着 $a=0$ ， $bb=255$ ），让 $cc$ 和 $dd$ 为在 $II$ 发现的最小和最大像素值。那么对比度拉伸后的图像 $OO$ 由函数给出。

$$OO(x, yy) = (II(x, yy) - cc) \diamond \frac{bb - a}{dd - cc} + a \quad (1)$$

问题1：写一个算法，按照上述公式（1）执行对比度拉伸。读取给定的灰度图像Kitten.png并运行你的算法，看看它是否真的

提高了图像质量。结果应该是这样的。

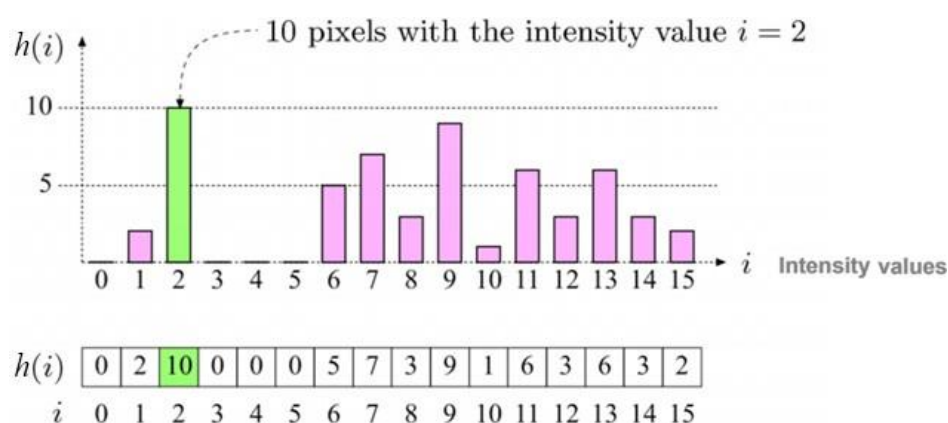


输入输出

还要写一个算法，找出图像中最小像素值的坐标和最大像素值的坐标。不要使用OpenCV的内置函数来完成这些任务，而要写你自己的代码。在输入图像和输出图像上运行它，并打印这些像素的值，以确认你的对比度拉伸算法是否正确工作。

## 2. 强度直方图

一个图像的直方图显示了强度值的计数。它只提供关于像素的统计信息，并删除了位置信息。对于一个数字图像来说，有  $LL$  灰度等级，从0到  $LL-1$ ，直方图是一个离散函数  $h(i) = n_{ii}$  其中  $i \in [0, LL-1]$  是  $i$  的灰度等级， $n_{ii}$  是该灰度等级的像素数。



**问题2：** 写一个算法来计算和绘制一个图像的直方图。不要使用OpenCV内置的函数来计算直方图，而要自己写代码来执行这个任务。然后在问题1中的Kitten.png和它的对比度拉伸版本上运行你的算法，并直观地比较直方图。

### 3. 图像边缘

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，不同强度的区域生活在不同的高度。图像中不同强度的区域之间的过渡意味着必须有一个陡峭的斜坡，我们将其正式称为梯度（矢量）。

$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad (2)$$

由于图像  $I$  是离散的，我们需要对连续导数  $\partial I / \partial x$  和  $\partial I / \partial y$  进行近似。

$\partial I / \partial y$  通过有限差分。进行有限差分的卷积核的简单例子是定义如下的 Sobel 滤波器。

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

**问题3：** 写一个算法，计算两个索贝尔图像  $\partial I / \partial x \approx I * S_x$  和

$\partial I / \partial y \approx I * S_y$  从一个输入图像。使用给定的图像 **CT.png** 来测试你的算法。不要使用 OpenCV 内置的函数来计算 Sobel 图像，而要写自己的代码来执行这个任务。你可以通过与内置函数的输出相比较来验证你自己算法的输出。

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。

之后，计算梯度大小的图像。

$$|\nabla I| = \sqrt{\left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2} \quad (3)$$

换句话说，创建一个新的输出图像，其大小与输入图像和 Sobel 图像相同。然后对于输出图像中的每个像素，计算其值为 Sobel 图像的平方值之和的平方根。 $\partial I / \partial x$  和 Sobel 图像的平方值  $\partial I / \partial y$  在该像素位置的值的平方根。

这里再次注意到，计算可能产生 8 位范围以外的中间值。因此，请确保你使用正确的数据类型进行计算。

最后的结果应该是这样的。



输入输出

#### 4. 图像构成

在科学和艺术的图像处理应用中，图像经常被组合起来，以创建一个具有特定属性的输出图像。这方面的例子已在图像处理第一部分讲座的最后一张幻灯片中展示。其中涉及的典型图像处理操作包括点运算（算术运算和逻辑运算）和邻域运算（图像过滤）。

**问题4（2.5分）。**写一个算法，执行以下图像处理步骤，以创建两个给定的输入图像**Cat.png**和**Dog.png**的组成图像。

1. 使用索贝尔滤镜对**Cat.png**进行边缘检测（我们把输出的图像称为**C**）。
2. 使用 $5 \times 5$ 的内核对**Dog.png**进行均匀滤波（让我们把输出的图像称为**DD**）。
3. 计算组成的图像  $00(x, yy) = 0.5 * C(x, yy) + 0.5 * D(x, y)$ 。

注意，输入的图像是RGB图像，所以上述步骤必须应用于每个通道，以产生组成的RGB输出图像，它应该看起来像这样。



猫狗合体

编码要求

对于所有的任务，要自己实现所需的算法，不要使用库函数。

从OpenCV（或任何其他软件包）中获得这些任务。使用这些函数而不是你的

自己执行将导致扣分。

在你的Jupyter笔记本中，输入图像应该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应该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

发布。2022年6月7日