

# 实验十报告

## 1.颜色直方图

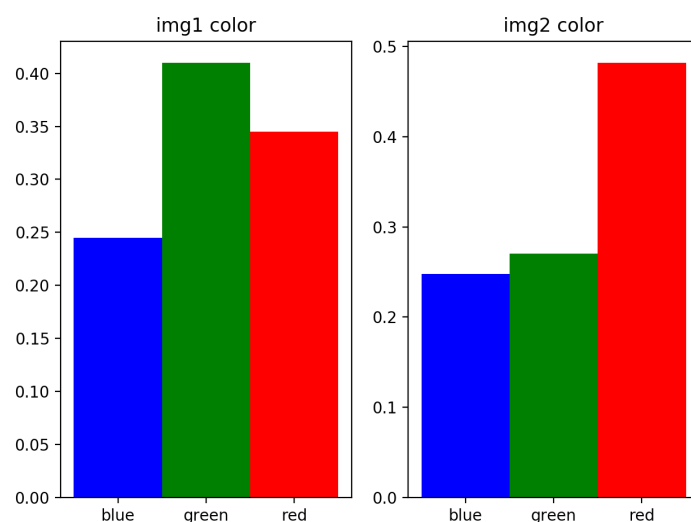
### 1.1.原理简述

颜色直方图是在许多图像检索系统中被广泛采用的颜色特征。它所描述的是不同色彩在整幅图像中所占的比例，而并不关心每种色彩所处的空间位置，即无法描述图像中的对象或物体。颜色直方图可以直观地展示图片的整体色调。

### 1.2.实现方法

首先使用opencv库的imread函数读取图片，读为RGB格式，即三维数组，分别为[B,G,R]。对每一层转换成numpy格式后，使用sum()函数统计图片不同颜色像素之和，再计算不同颜色像素所占比例。最后使用matplotlib画出直方图。

### 1.3.结果展示



## 2.灰度直方图

### 2.1.原理简述

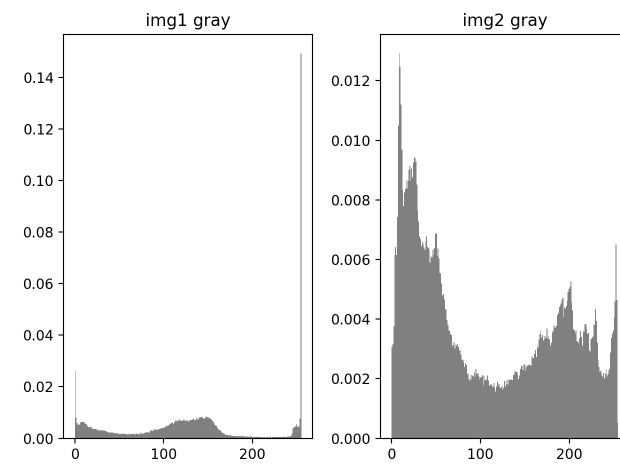
灰度直方图是关于灰度级分布的函数，是对图像中灰度级分布的统计。灰度直方图是将数字图像中的所有像素，按照灰度值的大小，统计其出现的频率。灰度直方图是灰度级的函数，它表示图像中具有某种灰度级的像素的个数，反映了图像中某种灰度出现的频率。

颜色直方图也可以直观地展示图片的整体明亮度。

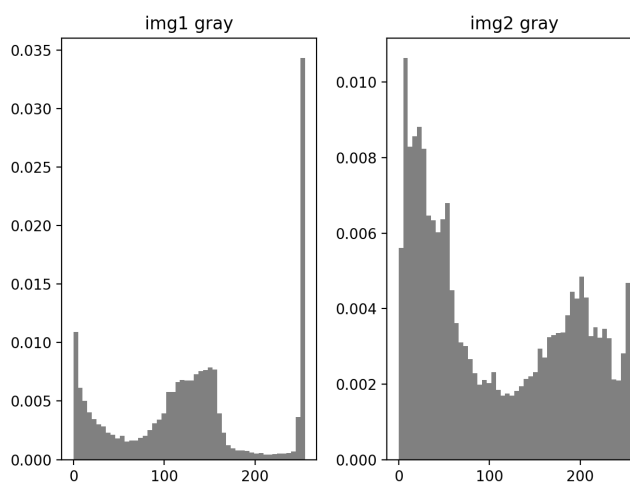
### 2.2.实现方法

首先使用opencv库的imread函数读取图片，读为灰度格式，仅一维矩阵。对这个矩阵采用numpy的reval()方法降维成一维，然后直接使用matplotlib库的hist函数画出直方图。需要注意的是，hist函数中设置normed为True以归一化，同时设置bins的个数，即将横坐标分为多少个组，聚类效果的精细度不同。

### 2.3.结果展示



设置bins=256



设置bins=50

### 3.梯度直方图

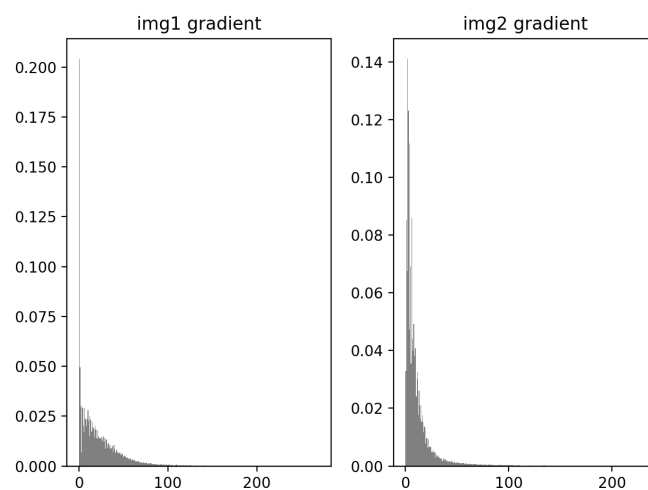
#### 3.1.原理简述

本次实验求得的是最原始的梯度直方图。即对于每个点，梯度强度 $=\sqrt{I_x^2 + I_y^2}$ ，而其中 $I_x, I_y$ 为其左右/上下的灰度值之差用来表示某方向上的梯度。

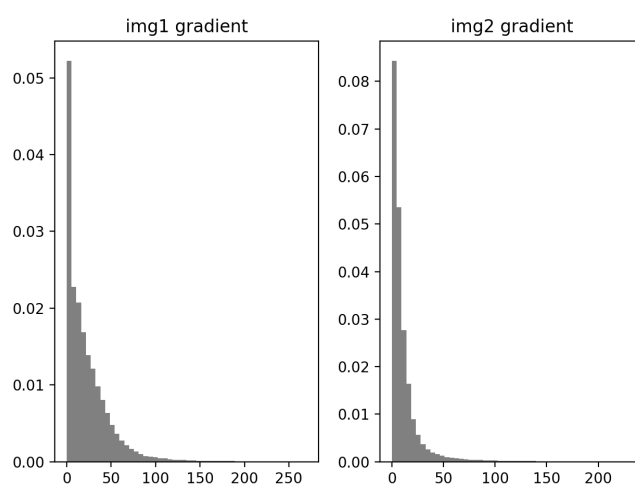
#### 2.2.实现方法

新建一个图层用来储存梯度。需要注意的是，这个图层的大小应该经过crop裁边，也就是确保其中每个点都有左右/上下的点。另外，不能直接对像素值进行运算，这样会越界255产生overflow；需要都强制类型转换到float后进行计算。另外，没有使用PPT中的向下取整的方法，可以直接用float值储存。

#### 2.3.结果展示



设置bins=361



设置bins=50