lab1-OpenCV　实验报告

电院 2303 徐恺阳 523030910085

November 2024

1　实验概览

本实验内容如下：

1. 了解数字图像在计算机中的表示和存储，包括灰度图像和彩色图像的表示；
2. 学习和使用 OpenCV 提取图像特征，包括颜色直方图、灰度直方图和梯度直方；
3. 熟悉 Matplotlib 绘制与保存图像。

2　实验环境

本实验基于 Python 3.10，用到的库有 OpenCV、Numpy 和 Matplotlib。

3　练习一

3.1　题目描述

将 img1.jpg、img2.jpg 和 img3.jpg 三幅图像以彩色图像方式读入，并计算颜色直方图。

3.2　解题思路

首先，分别以彩色图像和灰度图像的方式读取示例图像，代码如下：

```
img1 = cv2.imread('images/img1.jpg')
img1_rgb = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img1_gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)

img2 = cv2.imread('images/img2.jpg')
img2_rgb = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
img2_gray = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)

img3 = cv2.imread('images/img3.jpg')
img3_rgb = cv2.cvtColor(img3, cv2.COLOR_BGR2RGB)
img3_gray = cv2.cvtColor(img3, cv2.COLOR_BGR2GRAY)
```

图 1: 图像读取部分代码

然后，根据公式

$$E(c) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} I(x, y, c)$$

遍历 RGB 三个通道，计算每个通道的总能量。

$$H(c) = \frac{E(c)}{\sum_{i=0}^{2} E(i)}$$

将各通道的总能量归一化为比例，计算某一颜色分量的能量的相对比例。

代码如下：

```python
def calculate_color_energy(image_rgb):
    color_totals = [np.sum(image_rgb[..., i]) for i in range(3)]
    total_sum = sum(color_totals)
    normalized_totals = [x / total_sum for x in color_totals]
    return normalized_totals
```

图 2: 计算颜色分量相对比例部分代码

最后，根据结果绘制颜色直方图，代码如下：

```python
def plot_color_energy(ax, color_energy, title):
    colors = ['red', 'green', 'blue']
    ax.bar(range(3), color_energy, color=colors, tick_label=['R', 'G', 'B'])
    for i, v in enumerate(color_energy):
        ax.text(i, v + 0.02, f"{v:.2f}", ha='center')
    ax.set_ylim(0, 1)
    ax.set_title(title)
```

图 3: 绘制颜色直方图部分代码

3.3 代码运行结果

由于我是将练习一和练习二合在一起完成的，这里展示的是包含颜色直方图、灰度直方图和梯度直方图的一张图像。

注意：

- 这里按 PPT 要求，绘制的是每个颜色分量的平均比例，而非三个颜色分量的 0 ~ 255 每一个点的比例。

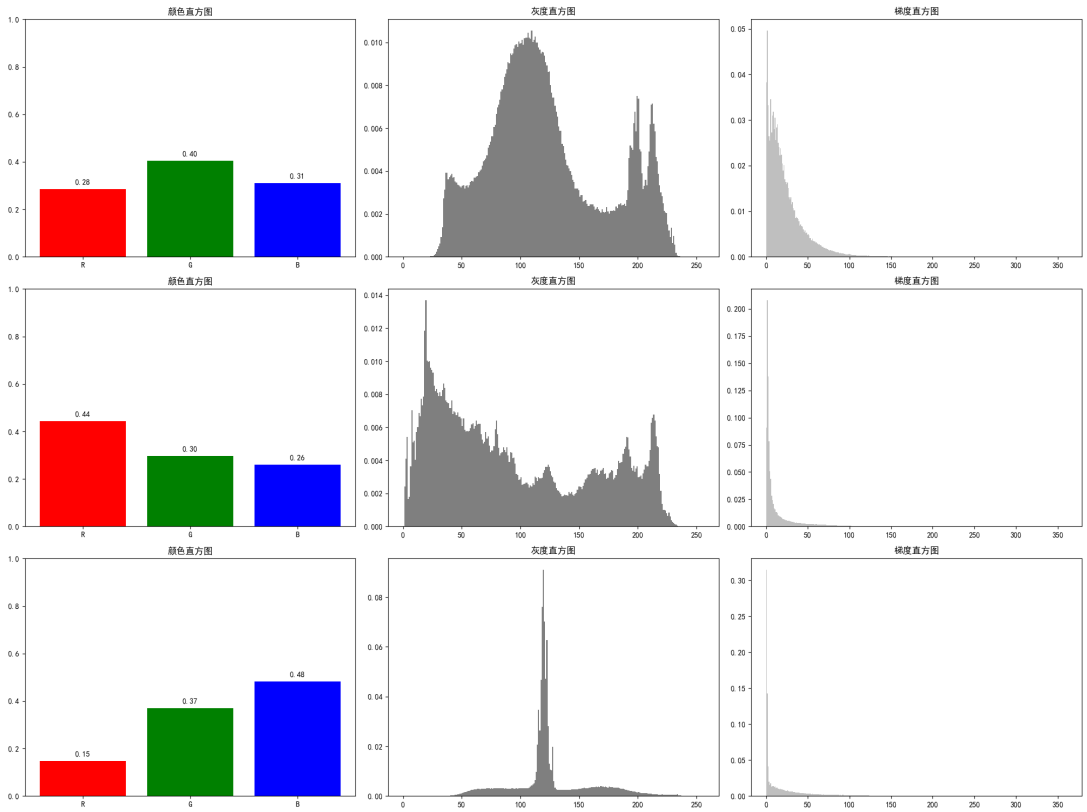

图 4: 颜色直方图、灰度直方图和梯度直方图

3.4 分析与思考

颜色直方图反映了图像的总体色调分布。例如在示例图片中，分别是绿色、红色、蓝色分量占主导，反映在颜色直方图中的结果就是绿色、红色、蓝色颜色分量的比例最高。

4 练习二

4.1 题目描述

将 img1.jpg、img2.jpg 和 img3.jpg 三幅图像以灰度图像方式读入，并计算灰度直方图和梯度直方图。

4.2 解题思路

练习一中已完成灰度图像的读入，这里不再赘述。

然后，根据公式

$$N(i) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (I(x,y) == i)?1 : 0$$

计算图像中灰度值为 i 的像素总个数。

$$H(i) = \frac{N(i)}{\sum_{j=0}^{255} N(j)}, \quad i = 0, \ldots, 255$$

计算灰度大小。

根据公式

$$N(i) = \sum_{x=1}^{W-2} \sum_{y=1}^{H-2} (B(x,y) == i)?1 : 0$$

计算像素值为 i 的像素总个数。

$$H(i) = \frac{N(i)}{\sum_{j=0}^{360} N(j)}$$

计算归一化后梯度大小。

代码如下：

```python
# 计算梯度图
def compute_gradient(image):
    # 将图像转换为浮点型以避免溢出
    image = image.astype(np.float32)

   # X方向梯度
    grad_x = np.zeros_like(image)
    grad_x[:, 1:-1] = image[:, 2:] - image[:, :-2]

    # Y方向梯度
    grad_y = np.zeros_like(image)
    grad_y[1:-1, :] = image[2:, :] - image[:-2, :]

    # 梯度幅值 M(x, y)
    gradient_magnitude = np.sqrt(grad_x**2 + grad_y**2)
    return gradient_magnitude[1:-1, 1:-1]
```

图 5: 计算梯度部分代码

最后，根据结果绘制灰度和梯度直方图，代码如下：

```python
def plot_histograms(image_rgb, image_gray, image_grad, ax_rgb_hist, ax_gray_hist, ax_grad_hist):

    color_energy1 = calculate_color_energy(image_rgb)
    plot_color_energy(ax_rgb_hist, color_energy1, "颜色直方图")

    # 灰度直方图
    ax_gray_hist.hist(image_gray.ravel(), bins=256, color='black', alpha=0.5, range=(0, 256), density=True)
    ax_gray_hist.set_title("灰度直方图")

    # 梯度直方图
    ax_grad_hist.hist(image_grad.ravel(), bins=361, color='gray', alpha=0.5, range=(0, 361), density=True)
    ax_grad_hist.set_title("梯度直方图")
```

图 6: 绘制灰度和梯度图部分代码

注意：

- 这里灰度和梯度图像均要求是频率分布直方图，故应选择 hist() 函数，并指定 density 参数为 True；
- 灰度直方图横轴范围（0，255），梯度图像横轴范围（0，360）；
- 计算梯度时不考虑最外围像素，故在计算梯度矩阵的时候应截取 [1:-1, 1:-1]。

4.3 代码运行结果

见图 4。

4.4 分析与思考

灰度直方图反映了图像明暗程度。例如在示例图片中，图片一灰度分布在中段区域，因此画面看起来较明亮；图片三灰度分布在较小的区域，因此画面看起来较黑暗。

梯度图反映了图像的纹理复杂程度。例如在示例图片中，图片一梯度分布较广，纹理较复杂；图片二、三中，梯度分布较窄，纹理简单。

5 实验感想

通过 lab1-OpenCV 实验，我了解了图像特征提取的基本方法，包括颜色直方图、灰度直方图和梯度直方图，并通过 OpenCV 的一些内置函数完成了对某一图像的的特征提取。同时，我也对 Matplotlib 的一些图像绘制与保存操作有了更进一步的了解。遇到的问题是不清楚 OpenCV 和 Matplotlib 提供的函数有哪些功能。

6 拓展思考

6.1 示例代码中的 cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB) 的作用是什么？

答：将 OpenCV 读取的图像颜色通道从 BGR（蓝-绿-红）顺序转换为 RGB（红-绿-蓝）顺序。

OpenCV 默认颜色通道顺序为 BGR，而 Matplotlib 默认颜色通道顺序为 RGB。如果直接将 OpenCV 读取的 BGR 图像传给 Matplotlib 显示，图像颜色会出现红色和蓝色互换的异常情况。

通过使用 cv2.cvtColor 函数将 BGR 转换为 RGB，可以确保图像在 Matplotlib 中以正确的颜色显示。

6.2 如何使得 pyplot 正确显示灰度图的颜色？

答：在 plt.imshow() 中指定 cmap 参数为'gray'。

cmap 参数，即 colormap，用于控制图像的颜色映射。显示灰度图时，需指定该参数为'gray'，即指定图像以灰度色阶显示。