《多传感器融合感知技术笔记》之
——图像处理灰度化/二值化_Akaxi

任务：利用 python 编写图像读取、显示图像，并基于 opencv 库函数和自定义函数实现图像灰度化、二值化。

图像定义为二维矩阵 A(u,v),其中，u,v 是像素坐标，a(i,j) 是图像在点（i,j）的处的强度或灰度的幅值。

彩色图像由三个（如 RGB）二维灰度（或亮度）矩阵 A(x,y)组成。用红、绿、蓝三元组的二维矩阵来表示。通常，三元组的 每个数值也是在 0 到 255 之间，0 表示相应的基色在该像素中没有，而 255 则代表相应的基色在该像素中取得最大值。

灰度图像是一个二维灰度（或亮度）矩 阵 A(i,j),每个像素的亮度用一个数值来表示，通常数值范围在 0 到 255 之间，0 表示黑、255 表示白，其它值表示处于黑白之间的灰度。

二值图像只有白和黑两种颜色，称为二值图像，0 表示黑，1 表示白。

一、使用 OpenCV 库函数实现图像灰度化、二值化

①读取彩色图像：使用函数 cv2.imread(filepath,flags)读入重邮和可莉图片

filepath：要读入图片的完整路径

flags：读入图片的标志

cv2.IMREAD_COLOR：默认参数，读入一副彩色图片，忽略 alpha 通道

cv2.IMREAD_GRAYSCALE：读入灰度图片

cv2.IMREAD_UNCHANGED：读入完整图片，包括 alpha 通道

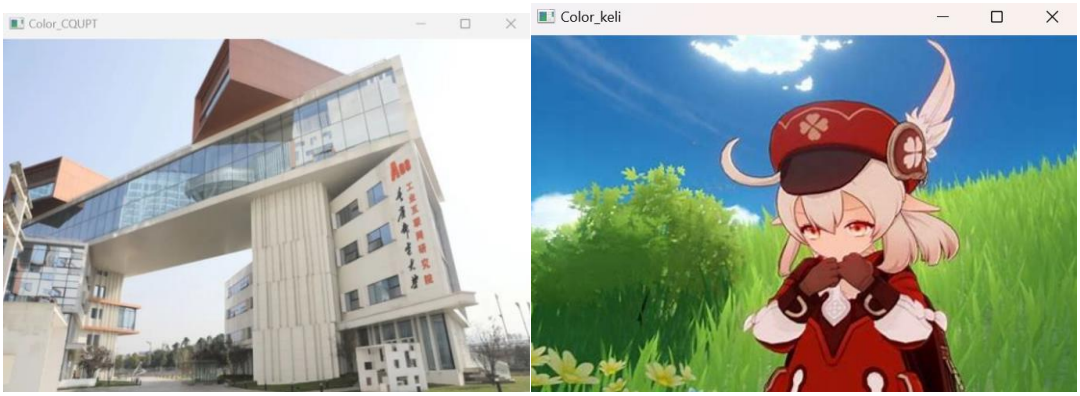

图 1 重邮（左）可莉（右）

②灰度化图像：使用 cv2.cvtColor(input_image, flag)函数灰度化图像

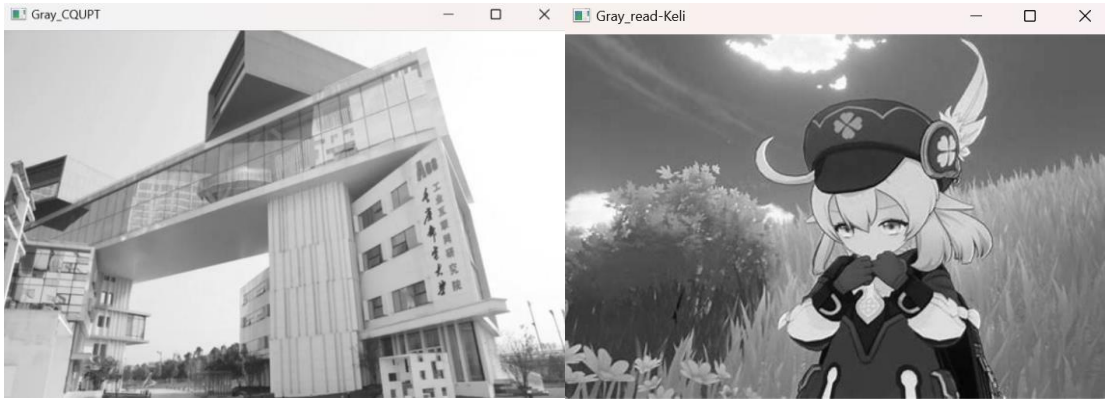

图 2 灰灰的重邮（左）灰灰的可莉（右）

```
cv2.cvtColor(input_image, flag)
```

参数	描述	返回值
input_image	需要转换的图片	颜色空间转换后的图片矩阵
flag	转换的类型	

转换格式：

类型	描述
cv2.COLOR_BGR2GRAY	BGR -> Gray
cv2.COLOR_BGR2RGB	BGR -> RGB
cv2.COLOR_BGR2HSV	BGR -> HSV

转换类型	Opencv2.x	Opencv3.x
RGB<-->BGR	CV_BGR2BGRA 、CV_RGB2BGRA 、CV_BGRA2RGBA 、CV_BGR2BGRA 、CV_BGRA2BGR	COLOR_BGR2BGRA,COLOR_RGB2BGRA 、COLOR_BGRA2RGBA,COLOR_BGR2BGRA 、COLOR_BGRA2BGR
RGB<-->GRAY	CV_RGB2GRAY 、CV_GRAY2RGB 、CV_RGBA2GRAY、CV_GRAY2GRBA	COLOR_RGB2GRAY,COLOR_GRAY2RGB 、COLOR_RGBA2GRAY,COLOR_GRAY2GRBA
RGB<-->HSV	CV_BGR2HSV 、CV_RGB2HSV 、CV_HSV2BGR、CV_HSV2RGB	COLOR_BGR2HSV、COLOR_RGB2HSV、COLOR_HSV2BGR、COLOR_HSV2RGB
RGB<-->YCrCb JPEG(或 YCC)	CV_RGB2YCrCb 、CV_RGB2YCrCb 、CV_YCrCb2BGR、CV_YCrCb2RGB (可以用 YUV 代替 YCrCb)	COLOR_RGB2YCrCb,COLOR_RGB2YCrCb, COLOR_YCrCb2BGR、COLOR_YCrCb2RGB (可以用 YUV 代替 YCrCb)
RGB <-->CIE XYZ	CV_BGR2XYZ,CV_RGB2XYZ, CV_XYZ2BGR, CV_XYZ2RGB	COLOR_BGR2XYZ,COLOR_RGB2XYZ, COLOR_XYZ2BGR, COLOR_XYZ2RGB
RGB<-->HLS	CV_BGR2HLS,CV_RGB2HLS, CV_HLS2BGR, CV_HLS2RGB	COLOR_BGR2HLS,COLOR_RGB2HLS, COLOR_HLS2BGR, COLOR_HLS2RGB
RGB<-->CIE L*a*b	CV_BGR2Lab,CV_RGB2Lab, CV_Lab2BGR, CV_Lab2RGB	COLOR_BGR2Lab,COLOR_RGB2Lab, COLOR_Lab2BGR, COLOR_Lab2RGB
RGB<-->CIE L*a*b	CV_BGR2Luv,CV_RGB2Luv, CV_Luv2BGR, CV_Luv2RGB	COLOR_BGR2Luv,COLOR_RGB2Luv, COLOR_Luv2BGR, COLOR_Luv2RGB

>或者在 imread 读取图像时直接读取成灰度图像,用 cv2.IMREAD_GRAYSCALE

③二值化图像：使用 cv2.threshold(src, thresh, maxval, type)函数二值化图像

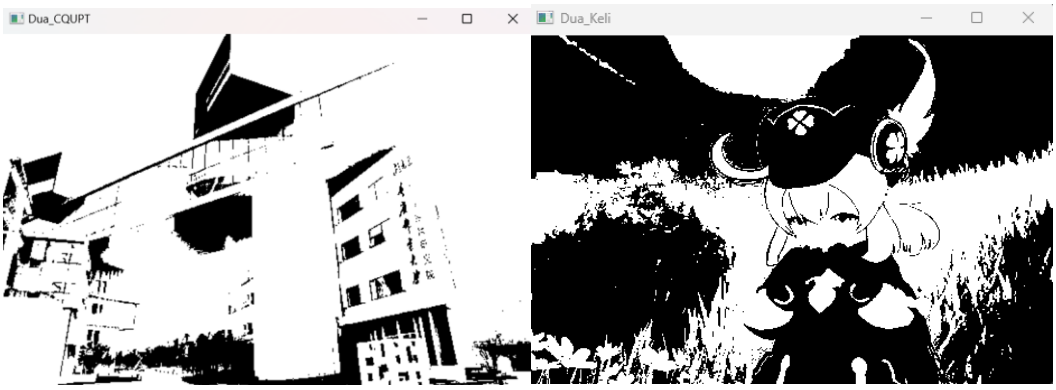

图 3 二值的重邮（左）二值的可莉（右）【thresh=127】

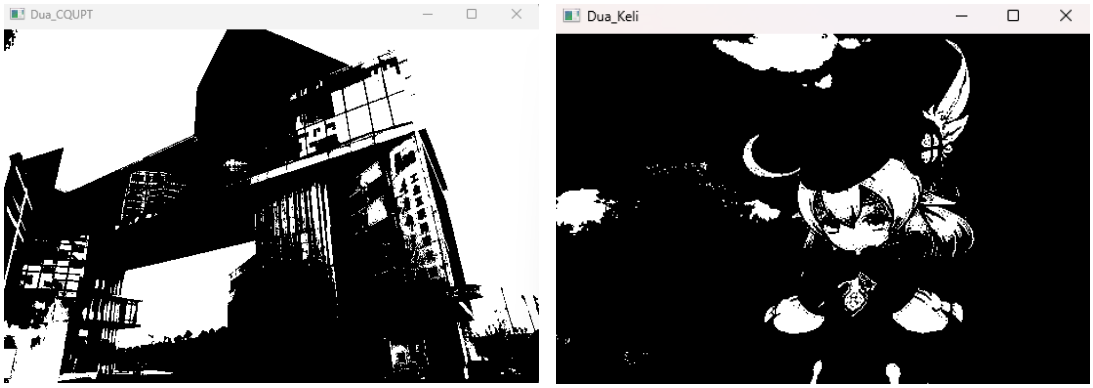

图 4 二值的重邮（左）二值的可莉（右）【thresh=200】

```
cv2.threshold(src, thresh, maxval, type)
```

参数	描述	返回
src	源图片，必须是单通道，即灰度图	返回两个值：阈值、二值图
thresh	用于对像素值进行分类的阈值，取值范围0～255	
maxval	填充色，如果像素值大于（有时小于）阈值则要给出的值，取值范围0～255	
type	阈值类型	

阈值类型表：

阈值	用数字表示	小于阈值的像素点	大于阈值的像素点
cv2.THRESH_BINARY	0	置0	置填充色maxval
cv2.THRESH_BINARY_INV	1	置填充色maxval	置0
cv2.THRESH_TRUNC	2	保持原色	置灰色
cv2.THRESH_TOZERO	3	置0	保持原色
cv2.THRESH_TOZERO_INV	4	保持原色	置0

④编写 python 源码

Picture_trans.py

```
1.      import cv2
2.      img_path = "C:/Users/Akaxi/Desktop/Akaxi_python/Sensor_learning_Akaxi/Keli.j
pg"  # 读取图片的路径
3.      img_path2 = "C:/Users/Akaxi/Desktop/Akaxi_python/Sensor_learning_Akaxi/fig1.
jpg"
4.
5.      # 使用 opencv 函数进行灰度化以及二值化
6.
7.      # imread 读取图像
8.      img_cv = cv2.imread(img_path)
9.      img_cv2 = cv2.imread(img_path2)
10.     cv2.imshow("Color_keli", img_cv)
11.     cv2.imshow("Color_CQUPT", img_cv2)
12.
13.     # 使用 cvtColor 函数灰度化图像
14.     img_cv_gray = cv2.cvtColor(img_cv, cv2.COLOR_BGR2GRAY)
15.     img_cv_gray2 = cv2.cvtColor(img_cv2, cv2.COLOR_BGR2GRAY)
16.     cv2.imshow("Gray_Keli", img_cv_gray)
17.     cv2.imshow("Gray_CQUPT", img_cv_gray2)
18.
19.     # 或者在 imread 读取图象时直接读取成灰度图像
20.     img_cv_st = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
21.     img_cv_st2 = cv2.imread(img_path2, cv2.IMREAD_GRAYSCALE)
22.     cv2.imshow("Gray_read-Keli", img_cv_st)
23.     cv2.imshow("Gray_read-CQUPT", img_cv_st2)
24.
25.     # 使用 threshold 函数二值化图像
26.     # ret, img_cv_dua = cv2.threshold(img_cv_gray, 127, 255, cv2.THRESH_BINARY)
27.     # ret2, img_cv_dua2 = cv2.threshold(img_cv_gray2, 127, 255, cv2.THRESH_BINAR
Y)
28.     ret, img_cv_dua = cv2.threshold(img_cv_gray, 200, 255, cv2.THRESH_BINARY)
29.     ret2, img_cv_dua2 = cv2.threshold(img_cv_gray2, 200, 255, cv2.THRESH_BINARY)
30.     cv2.imshow("Dua_Keli", img_cv_dua)
31.     cv2.imshow("Dua_CQUPT", img_cv_dua2)
32.     cv2.waitKey()
```

二、使用自定义函数实现图像灰度化、二值化

①读取彩色图像：使用函数 plt.imread()读入一副图片

图 5 重邮（左）可莉~（右）

②编写彩图转灰图函数：def rgb_to_gray(image):

功能：将 RGB 图像转换为灰度图像

参数：image:输入的 RGB 图像，类型为 NumPy 数组

灰度化思路：

先获取图片的尺寸，利用 np.shape 获取图片数组的长度和宽度，再创建同样大小的灰度图片 numpy 数组，使用循环来求取每一个像素灰度值,采用不同的算法，这里我采用均值法与经验函数法，核心函数为：

```
    for i in range(height):
        for j in range(width):
            # 均值法
            gray_value = (image[i, j, 0] + image[i, j, 1] + image[i, j, 2]) / 3
            # 加权平均法
            gray_value = 0.299 * image[i, j, 0] + 0.587 * image[i, j, 1] + 0.114 * image[i, j, 2]
            gray_img[i, j] = gray_value
```

还有其他方法，可以参考：

- 平均值法：将彩色图像中的三分量亮度求平均得到一个灰度图。
- 最大值法：将彩色图像中的三分量亮度的最大值作为灰度图的灰度值；
- 分量法：算法思想是先把RGB 每个分量的值作为图像的灰度值，这样就得到原图像的三个灰度图像，然后选择三个中的一个灰度图，即用RGB 三个分量的某一个分量作为该点的灰度值。
- 加权平均法：该算法主要就是根据某种条件，将三个分量以不同的权值进行加权平均；
  公式如下图所示：

分量法 $F1(i,j) = R(i,j)$ $F2(i,j) = G(i,j)$ $F3(i,j) = B(i,j)$

最大值法 $F(i,j) = \max(R(i,j), G(i,j), B(i,j))$

平均值法 $F(i,j) = (R(i,j) + G(i,j) + B(i,j)) / 3$

加权平均法 $F(i,j) = 0.30R(i,j) + 0.59G(i,j) + 0.11B(i,j))$

现以加权放大的灰度图变换为例，是将其三个分量以不同的权值进行加权平均，由于人眼对绿色敏感程度最高，对蓝色敏感最低，因此，按下式对RGB三分量进行加权平均能后得到较为合理的灰度图像。

$$F(i,j) = 0.30R(i,j) + 0.59G(i,j) + 0.11B(i,j))$$

③实例化得灰度图：

图 6 重邮 均值法（左）加权平均法（右）

图 7 可莉 均值法（左）加权平均法（右）

自己编写的函数使用不同方法得出的灰度图像效果不同，所以方法的选择还是比较重要。

其中在使用均值法时，会报错 RuntimeWarning: overflow encountered in ubyte_scalars，像素加减运算溢出异常，可能是由于在进行像素的 R、G、B 三通道值时累加会超出 255，所以报错，但是除以三均值后任然可以正常生成灰图。

④编写灰图转二值图函数：def gray_to_binary(gray_image, threshold):

功能：将灰度图像转换为二值图像

参数：gray_image：输入的灰度图像，类型为 NumPy 数组

threshold：二值化阈值，取值范围[0, 255]

二值化思路：

主要是将灰度值与阈值进行比较，0 或者 255 的二值化，核心代码为：

```
    for i in range(height):
        for j in range(width):
            if gray_image[i, j] > threshold:  # 如果像素的灰度值大于阈值，则为白色，否则为黑色
                binary_image[i, j] = 255
```

⑤实例化得二值图：

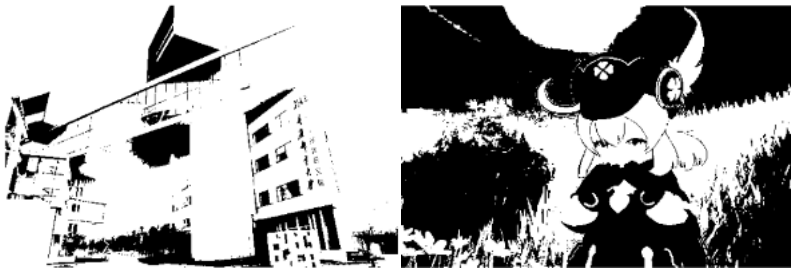

图 8 重邮（左）可莉（右）二值化【threshold=127】

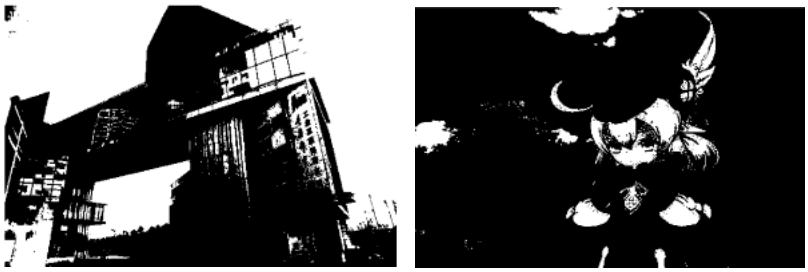

图 9 重邮（左）可莉（右）二值化【threshold=200】

⑥编写 python 源码

Picture_trans_own.py

```
1.      import numpy as np
2.      import matplotlib.pyplot as plt
3.
4.
5.      def rgb_to_gray(image):
6.          # 将 RGB 图像转换为灰度图像
7.          # image: 输入的 RGB 图像，类型为 NumPy 数组
8.          # 先获取图片的尺寸
9.          # shape 获取图片数组的长度和维度
10.         height, width, channels = image.shape
11.         gray_img = np.zeros((height, width), dtype=np.uint8)   # 创建同样大小的灰
度图片 numpy 数组
12.         # 使用循环来求取每一个像素灰度值,采用不同的算法
13.         for i in range(height):
14.             for j in range(width):
15.                 # gray_value = (image[i, j, 0] + image[i, j, 1] + image[i, j, 2]
) / 3    # 均值法
16.                 gray_value = 0.299 * image[i, j, 0] + 0.587 * image[i, j, 1] + 0
.114 * image[i, j, 2]  # 加权平均法
17.                 gray_img[i, j] = gray_value
18.
19.         return gray_img
20.
21.
22.     def gray_to_binary(gray_image, threshold):
23.         # 将灰度图像转换为二值图像
24.         # gray_image: 输入的灰度图像，类型为 NumPy 数组
25.         # threshold: 二值化阈值，取值范围[0, 255]
26.
27.         # 获取图像的宽度和高度
28.         height, width = gray_image.shape
29.         binary_image = np.zeros((height, width), dtype=np.uint8)  # 创建同样大小
的灰度图片 numpy 数组
30.         # 使用循环来对每一个像素进行二值化处理
31.         for i in range(height):
32.             for j in range(width):
```

```
33.                 if gray_image[i, j] > threshold:  # 如果像素的灰度值大于阈值，则为白色，否则为黑色
34.                     binary_image[i, j] = 255
35.
36.         return binary_image
37.
38.     # 【灰度化】
39.     # 读取图像 matplotlib 获取的图片会变成 numpy 的数组
40.     image = plt.imread('C:/Users/Akaxi/Desktop/Akaxi_python/Sensor_learning_Akaxi/Keli.jpg')
41.     image2 = plt.imread('C:/Users/Akaxi/Desktop/Akaxi_python/Sensor_learning_Akaxi/fig1.jpg')
42.
43.     # 显示彩色图
44.     plt.imshow(image)
45.     plt.axis('off')
46.     plt.show()
47.
48.     plt.imshow(image2)
49.     plt.axis('off')
50.     plt.show()
51.
52.     # 将彩色图像转换为灰度图像
53.     gray_img = rgb_to_gray(image)
54.     gray_img2 = rgb_to_gray(image2)
55.
56.     # 显示灰度图
57.     plt.imshow(gray_img, cmap='gray')
58.     plt.axis('off')
59.     plt.show()
60.
61.     plt.imshow(gray_img2, cmap='gray')
62.     plt.axis('off')
63.     plt.show()
64.
65.     # 【二值化】
66.     # 加载灰度图
67.     gray_image = gray_img
68.     gray_image2 = gray_img2
69.
70.     # 将灰度图像转换为二值图
71.     threshold = 127  # 二值化阈值
```

```
    # threshold = 200  # 二值化阈值
    binary_image = gray_to_binary(gray_image, threshold)
    binary_image2 = gray_to_binary(gray_image2, threshold)

    # 显示二值图
    plt.imshow(binary_image, cmap='gray')
    plt.axis('off')
    plt.show()

    plt.imshow(binary_image2, cmap='gray')
    plt.axis('off')
    plt.show()
```


[1]cv2.cvtColor(input_image, flag)参考博文

https://blog.csdn.net/weixin_40522801/article/details/106517099

[2]cv2.threshold(src, thresh, maxval, type)参考博文

https://blog.csdn.net/weixin_40522801/article/details/106516424

[3]自己编写灰度图参考博文

https://blog.csdn.net/rhyijg/article/details/106934061

[4] 自己编写二值图参考博文

https://blog.csdn.net/bblingbbling/article/details/112793681

2023.9.13

渝北仙桃