

图像处理与分析第一次作业

162050127 颜劭铭

2022 年 10 月 2 日

1 计算并显示以下图像的直方图

矩阵操作进行计算

```
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv

def imghistogram(img):
    """
    根据图像绘制直方图
    """
    h, w = img.shape # 获取图像的高与宽
    hist = [0] * 256 # 首先各灰度级频数都置为0
    for i in range(h):
        for j in range(w):
            hist[img[i, j]] += 1 # 矩阵操作
    return hist

img = cv.imread('1.png', 0) # 以灰度图像读入
plt.bar(range(256), imghistogram(img), width=1)
plt.grid(True, linestyle=':', color='r', alpha=0.6)
plt.title("Gray histogram")
plt.show()
```

图 1: 代码

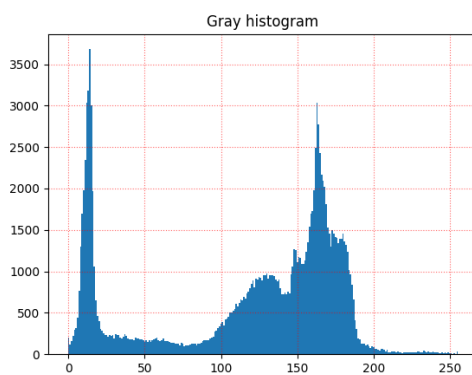


图 2: 结果

2 2. 分别对以下图片进行图像增强

2.1 第一张

将 RGB 拆成 R,G,B 三个通道矩阵，分别对 3 通道进行直方图均衡化，然后合成为 RGB 图

```
import numpy as np
import cv2 as cv

img = cv.imread("2.png")
B,G,R = cv.split(img)
BH = cv.equalizeHist(B) #分别对3通道进行直方图均衡化
GH = cv.equalizeHist(G)
RH = cv.equalizeHist(R)
result = cv.merge((BH,GH,RH),)#通道合成
res = np.hstack((img,result))

cv.imwrite("after2.png", res)
```

图 3: 代码



图 4: 结果

2.2 第二张

将 RGB 拆成 R,G,B 三个通道矩阵，分别对 3 通道进行直方图均衡化，然后合成为 RGB 图

```

import numpy as np
import cv2 as cv

img = cv.imread("3.png")
B,G,R = cv.split(img)
BH = cv.equalizeHist(B) #分别对3通道进行直方图均衡化
GH = cv.equalizeHist(G)
RH = cv.equalizeHist(R)
result = cv.merge((BH,GH,RH),)#通道合成
res = np.hstack([img,result])

cv.imwrite("after3.png", res)

```

图 5: 代码



图 6: 结果

2.3 第三张

中值滤波，设置了不同滤波器大小，滤波器大小为 3×3 的时候效果最好，噪点最少且不模糊

```

def median_filter(x, y, step):
    ... sum_s=[] # 定义空数组
    ... for k in range(-int(step/2), int(step/2)+1):
    ...     for m in range(-int(step/2), int(step/2)+1):
    ...         sum_s.append(img[x+k][y+m]) # 把模块的像素添加到空数组
    ... sum_s.sort() # 对模板的像素由小到大进行排序
    ... return sum_s[(int(step*step/2)+1)]

MedStep = [2,3,4,5,6] #设置滤波器

for k in range(0,5):
    ... medstep = MedStep[k]
    ... for i in range(int(medstep/2), h - int(medstep/2)):
    ...     for j in range(int(medstep/2), w - int(medstep/2)):
    ...         if(medstep == 2):
    ...             img_copytwo[i][j] = median_filter(i, j, medstep)
    ...         elif(medstep == 3):
    ...             img_copythree[i][j] = median_filter(i, j, medstep)
    ...         elif(medstep == 4):
    ...             img_copyfour[i][j] = median_filter(i, j, medstep)
    ...         elif(medstep == 5):
    ...             img_copyfive[i][j] = median_filter(i, j, medstep)
    ...         elif(medstep == 6):
    ...             img_copysix[i][j] = median_filter(i, j, medstep)

#创建一个窗口
plt.figure('contrast',figsize=(7,5))
#显示原图
plt.subplot(321) #子图1
plt.imshow(img,plt.cm.gray)
plt.title('before')

#显示处理过的图像

plt.subplot(322) #子图2
plt.imshow(img_copytwo,plt.cm.gray)
plt.title('filter_two')

```

图 7: 代码

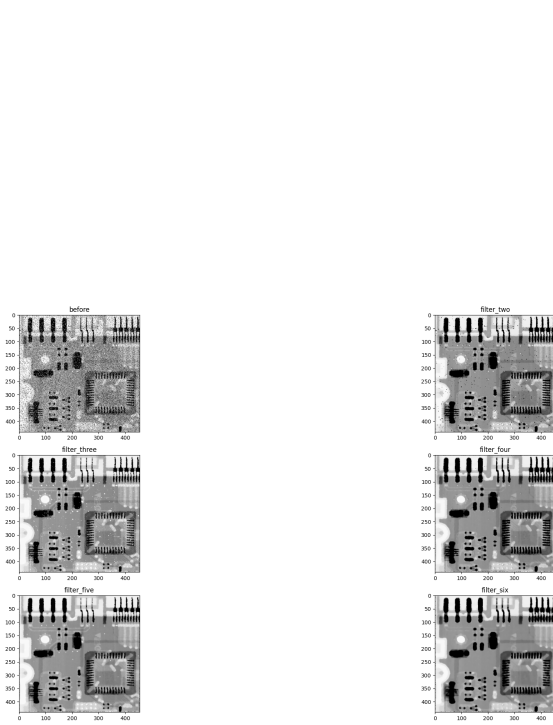


图 8: 结果