

图像直方图均衡化实验

班级： 21计科4班 姓名： 陈昊天

一、实验目的

- 1、掌握用python 语言对图像的读写操作；
- 2、掌握图像直方图增强（直方图均衡化）算法；
- 3、能够分析实验结果对算法进行评估。

二、实验原理

图像直方图均衡化是一种图像增强技术，主要用于改善图像的对比度。该方法的基本思想是将原始图像的灰度值分布通过一定的变换，使得输出图像的灰度值在直方图中更均匀地分布，从而增强图像的整体对比效果，突出图像中的细节信息。

直方图均衡化的步骤包括：

- (1) 计算图像的直方图：统计图像中每个灰度级别（0-255）出现的频率。
- (2) 计算累计分布函数CDF：将每个灰度级别的频率累加，得到每个灰度级的累计概率。
- (3) 归一化CDF并映射像素值：将归一化后的CDF乘以最大灰度值255，再对图像中的每个像素进行灰度值映射，得到新的像素值。
- (4) 生成增强后的图像：通过映射后的灰度值，重新构建图像，达到增强效果。

该方法对于背景和前景都比较暗或都比较亮的图像尤为有效，可以显著提高图像的视觉效果。

三、实验环境

开发语言：python

下载网址：<https://www.anaconda.com/download/success>

从以上网址下载anaconda，并安装spyder，安装opencv库

本实验所需要库：Numpy 库、Matplotlib 库和 OpenCV2 库。Numpy 库、CV2库、Math 库、matplotlib 库和 Pandas 库。Numpy 库主要用于对多维数组执行计算，极大地简化了向量和矩阵的操作处理；OpenCV2（OpenCV 是一个基于 BSD 许可(开源)发行的跨平台计算机视觉库）；Matplotlib 是 Python 中最常用的可视化工具之一，可以非常方便地创建海量类型地 2D 图表和一些基本的 3D 图表，绘制图形（线形图，柱状图，直方图，密度图，散布图等等）。skimage 即是 Scikit-Image，是基于 python 脚本语言开发的数字图片处理包

安装库指令：

```
pip install  
numpy pip  
install  
matplotlib  
pip install opencv-  
python pip install  
scikit-image
```

导入库和模块指令:

```
import cv2  
from matplotlib import pyplot  
as plt import numpy as np  
import math  
import pandas as pd  
from skimage import metrics, draw
```

四、实验代码与运行结果

实验代码

```
import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif'] = ['Arial Unicode MS']  
plt.rcParams['axes.unicode_minus'] = False  
  
def histogram_equalization(image):  
    height, width = image.shape  
    pixel_count = height * width  
  
    histogram = np.zeros(256)  
    for i in range(height):  
        for j in range(width):  
            histogram[image[i,j]] += 1  
  
    cdf = np.zeros(256)  
    cdf[0] = histogram[0]  
    for i in range(1, 256):
```

```

        cdf[i] = cdf[i-1] + histogram[i]

    cdf_normalized = (cdf - cdf.min()) * 255 / (pixel_count -
cdf.min())
    cdf_normalized = cdf_normalized.astype(np.uint8)

    equalized_image = np.zeros((height, width), dtype=np.uint8)
    for i in range(height):
        for j in range(width):
            equalized_image[i,j] = cdf_normalized[image[i,j]]

    return equalized_image

def plot_histogram(image, title):
    plt.figure(figsize=(8, 4))
    plt.hist(image.ravel(), 256, [0, 256])
    plt.title(title)
    plt.xlabel('像素值')
    plt.ylabel('频率')

image = cv2.imread('input.jpg', cv2.IMREAD_GRAYSCALE)

equalized_image = histogram_equalization(image)

plt.figure(figsize=(12, 4))

plt.subplot(121)
plt.imshow(image, cmap='gray')
plt.title('原始图像')
plt.axis('off')

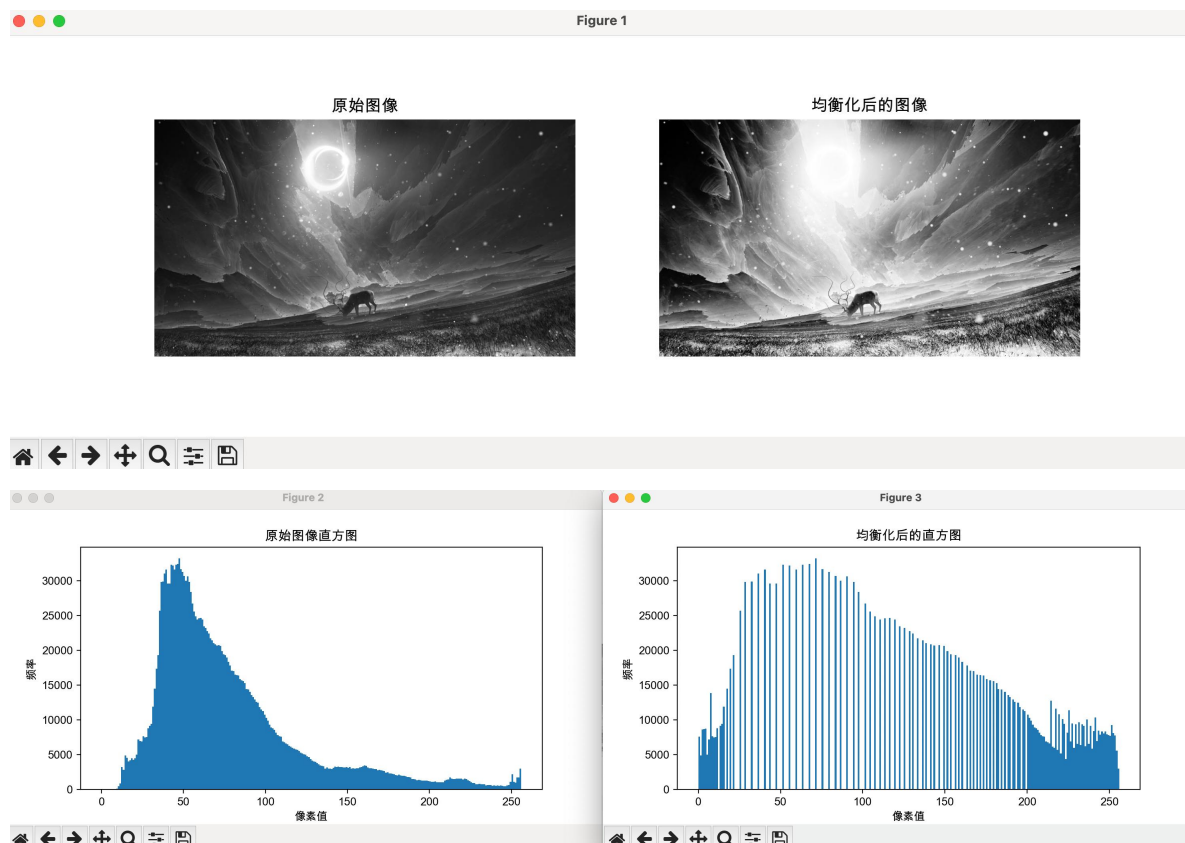
plt.subplot(122)
plt.imshow(equalized_image, cmap='gray')
plt.title('均衡化后的图像')
plt.axis('off')

plot_histogram(image, '原始图像直方图')
plot_histogram(equalized_image, '均衡化后的直方图')

```

```
cv2.imwrite('original.jpg', image)
cv2.imwrite('equalized.jpg', equalized_image)
```

运行结果



五、结果分析与心得体会

本次实验我们成功实现图像直方图均衡化的算法，并使用 Python 中的 OpenCV 和 Matplotlib 库对图像进行了处理和可视化展示。从实验结果来看，图像经过直方图均衡化处理后，灰度值分布变得更加均匀，图像的对比度得到了显著提升，原本较为模糊的细节部分变得更加清晰可见。

实验中，通过绘制图像的直方图，可以直观地看到原始图像灰度值集中分布，而均衡化后的图像灰度值分布更广泛、趋于平衡，验证算法的有效性。

通过本实验加深我对图像处理基本原理的理解，提升使用 Python 进行图像编程的能力。在实践过程中，我也体会到算法实现与可视化的重要性，能够更好地理解和分析图像增强的效果。