# 实验一.直方图均衡化

## 代码分析

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @FilePath     : \Opencv\_Study\Z\_ClassTask\task1.c

\* @Author       : Yang Shuaige

\* @File Version : V1.0.0

\* @Create Date  : 2024-03-18 10:03:04

\* @Description  : 直方图均衡化

\* @Copyright (c) 2024 by Yang Shuaige, All Rights Reserved.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <opencv2/highgui.hpp>

#include <opencv2/imgproc.hpp>

//#include <opencv2/matplotlib/matplotlib/matplotlibcpp.h>  //绘图函数库，绘制图片对应直方图

#include <iostream>

using namespace cv;

int main(int *argc*, char \**argv*[])

{

    Mat image = imread("F:\\Deep Learning\\Opencv\_Study\\material\\lena.jpg",0);//给定参数0读入灰度图

   //图像获取失败

    if(image.empty())

    {

        std::cout << "Could not open or find the image" << std::endl;

        return -1;

    }

    imshow("原图",image);

    //此部分代码是将BGR三种基础色分别进行均衡化的操作，通过split将图片的像素分为三维数组，然后分别进行equalizeHist操作。

    Mat imageRGB[3];

    split(image,imageRGB);

    for(int i=0;i<3;i++)

    {

        equalizeHist(imageRGB[i],imageRGB[i]);

    }

     merge(imageRGB,3,image);

    //只进行灰度图的直方图均衡化

    //equalizeHist(image,image);

    imshow("直方图均衡化",image);

    waitKey(0);

    return 0;

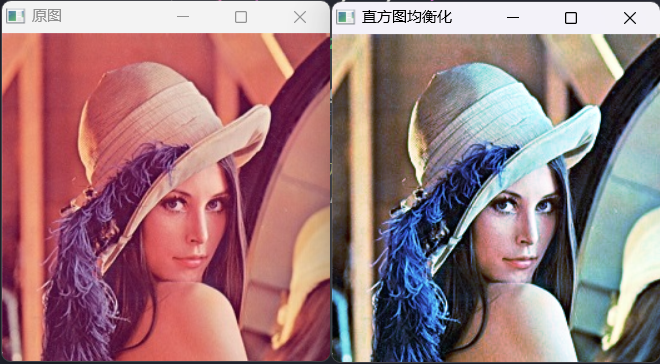
}

代码执行思路

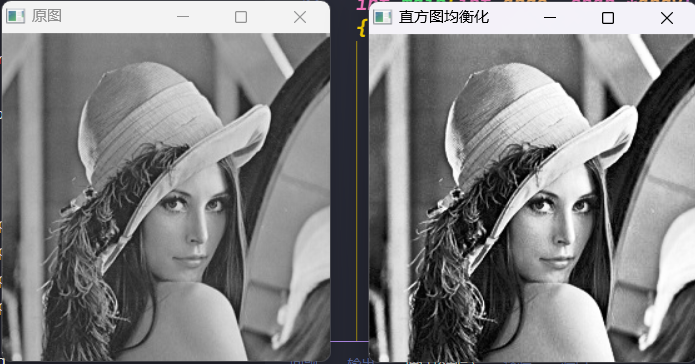
* 通过imread读取图像，默认是彩色图像，如果想获取单色调的图像，需要改变imread的第二个参数
* Imshow()显示出原图
* 通过split将彩色图片的BGR各自对应元素分开，然后通过循环分别进行直方图均衡化
* 将均衡化后的BGR再通过merge结合到一块
* 显示均衡化后的图片

## 结果分析

* 1. 彩图C++均衡化结果



* 1. 灰度图C++均衡化结果



在进行灰度图的均衡化时只需要将如下代码注释掉

//此部分代码是将BGR三种基础色分别进行均衡化的操作，通过split将图片的像素分为三维数组，然后分别进行equalizeHist操作。

    // Mat imageRGB[3];

    // split(image,imageRGB);

    // for(int i=0;i<3;i++)

    // {

    //     equalizeHist(imageRGB[i],imageRGB[i]);

    // }

    //  merge(imageRGB,3,image);

直接使用

equalizeHist(image,image);

即可进行均衡化操作。注意读入图像要改为灰度图

## 结果可视化

* 1. **Python代码**

import cv2

import numpy as np

from matplotlib import pyplot as plt

#显示图像函数

def cv\_show(*image*,*img*):

    cv2.imshow(*image*,*img*)

    cv2.waitKey(0)

    cv2.destroyAllWindows()

#读取图像

img = cv2.imread('F:/Deep Learning/Opencv\_Study/material/lena.jpg',0)

plt.subplot(121)

plt.hist(img.ravel(),256)

cv\_show('img',img)

equ = cv2.equalizeHist(img)

cv\_show('equ',equ)

plt.subplot(121)

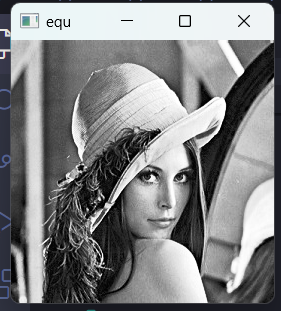
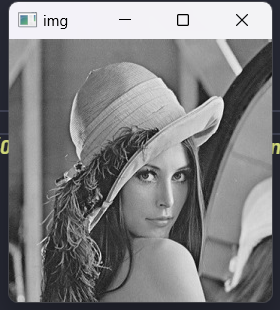
plt.hist(img.ravel(),256)

plt.subplot(122)

plt.hist(equ.ravel(),256)

**plt.show()**

* 1. **Python代码灰度图均衡化结果**



结果与C++实现结果无差别

* 1. **均衡化前后直方图可视化**

