

该系列文章是讲解Python OpenCV图像处理知识，前期主要讲解图像入门、OpenCV基础用法，中期讲解图像处理的各种算法，包括图像锐化算子、图像增强技术、图像分割等，后期结合深度学习研究图像识别、图像分类应用。希望文章对您有所帮助，如果有不足之处，还请海涵~

该系列在github所有源代码：<https://github.com/eastmountyxz/ImageProcessing-Python>
PS：请求帮忙点个Star，哈哈，第一次使用Github，以后会分享更多代码，一起加油。

同时推荐作者的C++图像系列知识：

[数字图像处理] 一.MFC详解显示BMP格式图片

[数字图像处理] 二.MFC单文档分割窗口显示图片

[数字图像处理] 三.MFC实现图像灰度、采样和量化功能详解

[数字图像处理] 四.MFC对话框绘制灰度直方图

[数字图像处理] 五.MFC图像点运算之灰度线性变化、灰度非线性变化、阈值化和均衡化处理详解

[数字图像处理] 六.MFC空间几何变换之图像平移、镜像、旋转、缩放详解

[数字图像处理] 七.MFC图像增强之图像普通平滑、高斯平滑、Laplacian、Sobel、Prewitt锐化详解

前文参考：

[Python图像处理] 一.图像处理基础知识及OpenCV入门函数

[Python图像处理] 二.OpenCV+Numpy库读取与修改像素

[Python图像处理] 三.获取图像属性、兴趣ROI区域及通道处理

[Python图像处理] 四.图像平滑之均值滤波、方框滤波、高斯滤波及中值滤波

[Python图像处理] 五.图像融合、加法运算及图像类型转换

[Python图像处理] 六.图像缩放、图像旋转、图像翻转与图像平移

[Python图像处理] 七.图像阈值化处理及算法对比

[Python图像处理] 八.图像腐蚀与图像膨胀

[Python图像处理] 九.形态学之图像开运算、闭运算、梯度运算

[Python图像处理] 十.形态学之图像顶帽运算和黑帽运算

[Python图像处理] 十一.灰度直方图概念及OpenCV绘制直方图

[Python图像处理] 十二.图像几何变换之图像仿射变换、图像透视变换和图像校正

[Python图像处理] 十三.基于灰度三维图的图像顶帽运算和黑帽运算

[Python图像处理] 十四.基于OpenCV和像素处理的图像灰度化处理

前一篇文章讲解了图像灰度化处理的知识，结合OpenCV调用cv2.cvtColor()函数实现图像灰度操作，使用像素处理方法对图像进行灰度化处理。本文主要讲解灰度线性变换，基础性知识希望对您有所帮助。

1.图像灰度上移变换： $DB=DA+50$

2.图像对比度增强变换： $DB=DA*1.5$

3.图像对比度减弱变换: $DB=DA*0.8$

4.图像灰度反色变换: $DB=255-DA$

PS: 文章参考自己以前系列图像处理文章及OpenCV库函数, 同时参考如下文献:

杨秀璋等. 基于苗族服饰的图像锐化和边缘提取技术研究[J]. 现代计算机, 2018(10).

《数字图像处理》(第3版), 冈萨雷斯著, 阮秋琦译, 电子工业出版社, 2013年.

《数字图像处理学》(第3版), 阮秋琦, 电子工业出版社, 2008年, 北京.

《OpenCV3编程入门》, 毛星云, 冷雪飞, 电子工业出版社, 2015.

[数字图像处理] 五.MFC图像点运算之灰度线性变化、灰度非线性变化、阈值化和均衡化处理详解

一.图像灰度线性变换原理

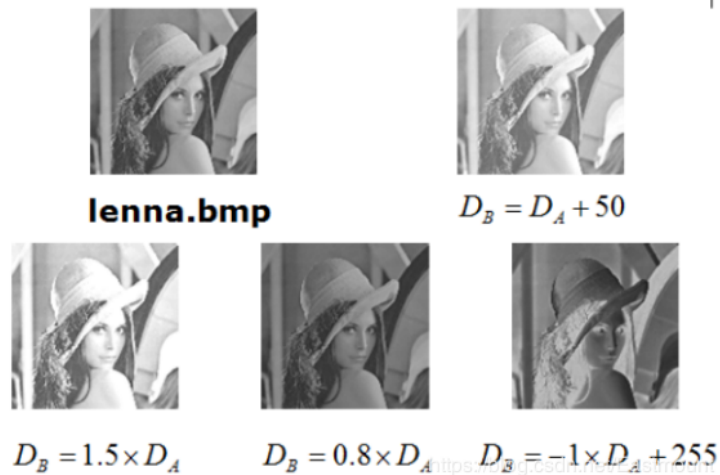
图像的灰度线性变换是通过建立灰度映射来调整原始图像的灰度, 从而改善图像的质量, 凸显图像的细节, 提高图像的对比度。灰度线性变换的计算公式如下所示:

$$D_B = f(D_A) = \alpha D_A + b$$

该公式中 D_B 表示灰度线性变换后的灰度值, D_A 表示变换前输入图像的灰度值, α 和 b 为线性变换方程 $f(D)$ 的参数, 分别表示斜率和截距。

- 当 $\alpha=1$, $b=0$ 时, 保持原始图像
- 当 $\alpha=1$, $b \neq 0$ 时, 图像所有的灰度值上移或下移
- 当 $\alpha=-1$, $b=255$ 时, 原始图像的灰度值反转
- 当 $\alpha>1$ 时, 输出图像的对比度增强
- 当 $0<\alpha<1$ 时, 输出图像的对比度减小
- 当 $\alpha<0$ 时, 原始图像暗区域变亮, 亮区域变暗, 图像求补

如图所示, 显示了图像的灰度线性变换对应的效果图。



二.图像灰度上移变换

该算法将实现图像灰度值的上移，从而提升图像的亮度，其实现代码如下所示。由于图像的灰度值位于0至255区间之内，所以需要对灰度值进行溢出判断。

```
# -*- coding: utf-8 -*-
import cv2
import numpy as np
import matplotlib.pyplot as plt

# 读取原始图像
img = cv2.imread('miao.png')

# 图像灰度转换
grayImage = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 获取图像高度和宽度
height = grayImage.shape[0]
width = grayImage.shape[1]

# 创建一幅图像
result = np.zeros((height, width), np.uint8)

# 图像灰度上移变换 DB=DA+50
for i in range(height):
    for j in range(width):

        if (int(grayImage[i,j]+50) > 255):
            gray = 255
        else:
```

```
gray = int(grayImage[i,j]+50)

result[i,j] = np.uint8(gray)

#显示图像
cv2.imshow("Gray Image", grayImage)
cv2.imshow("Result", result)

#等待显示
cv2.waitKey(0)
cv2.destroyAllWindows()
```

其输出结果如下图所示，图像的所有灰度值上移50，图像变得更白了。注意，纯黑色对应的灰度值为0，纯白色对应的灰度值为255。



三.图像对比度增强变换

该算法将增强图像的对比度，Python实现代码如下所示：

```
# -*- coding: utf-8 -*-
import cv2
import numpy as np
import matplotlib.pyplot as plt

#读取原始图像
img = cv2.imread('miao.png')

#图像灰度转换
grayImage = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
# 获取图像高度和宽度
height = grayImage.shape[0]
width = grayImage.shape[1]

# 创建一幅图像
result = np.zeros((height, width), np.uint8)

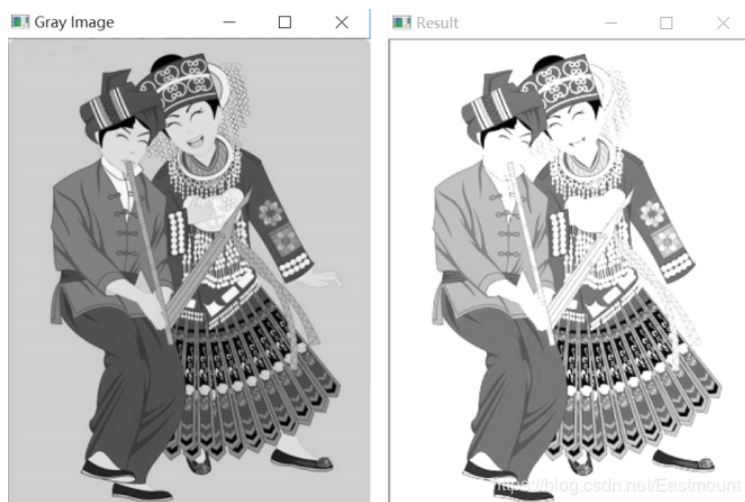
# 图像对比度增强变换  $DB=DA*1.5$ 
for i in range(height):
    for j in range(width):

        if (int(grayImage[i,j]*1.5) > 255):
            gray = 255
        else:
            gray = int(grayImage[i,j]*1.5)

        result[i,j] = np.uint8(gray)

# 显示图像
cv2.imshow("Gray Image", grayImage)
cv2.imshow("Result", result)
```

其输出结果如下图所示，图像的所有灰度值增强1.5倍。



四.图像对比度减弱变换

该算法将减弱图像的对比度，Python实现代码如下所示：

```
# -*- coding: utf-8 -*-
import cv2
```

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

# 读取原始图像
img = cv2.imread('miao.png')

# 图像灰度转换
grayImage = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 获取图像高度和宽度
height = grayImage.shape[0]
width = grayImage.shape[1]

# 创建一幅图像
result = np.zeros((height, width), np.uint8)

# 图像对比度减弱变换  $DB=DA*0.8$ 
for i in range(height):
    for j in range(width):
        gray = int(grayImage[i,j]*0.8)
        result[i,j] = np.uint8(gray)

# 显示图像
cv2.imshow("Gray Image", grayImage)
cv2.imshow("Result", result)

# 等待显示
cv2.waitKey(0)
cv2.destroyAllWindows()
```

其输出结果如下图所示，图像的所有灰度值减弱，图像变得更暗。



五.图像灰度反色变换

反色变换又称为线性灰度求补变换，它是对原图像的像素值进行反转，即黑色变为白色，白色变为黑色的过程。其Python实现代码如下所示：

```
# -*- coding: utf-8 -*-
import cv2
import numpy as np
import matplotlib.pyplot as plt

# 读取原始图像
img = cv2.imread('miao.png')

# 图像灰度转换
grayImage = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 获取图像高度和宽度
height = grayImage.shape[0]
width = grayImage.shape[1]

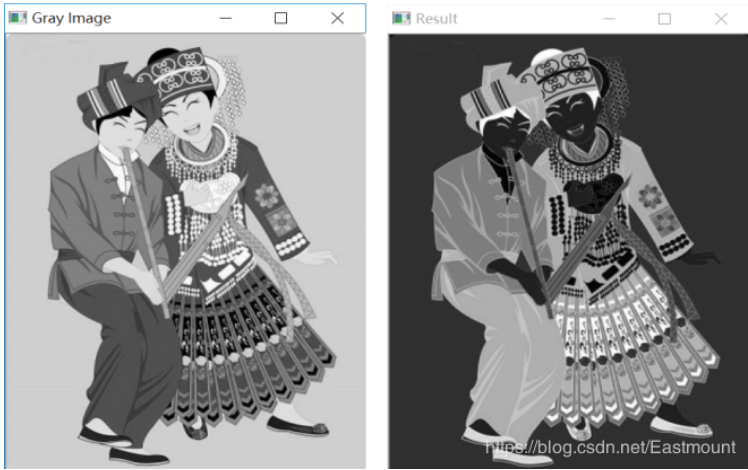
# 创建一幅图像
result = np.zeros((height, width), np.uint8)

# 图像灰度反色变换 DB=255-DA
for i in range(height):
    for j in range(width):
        gray = 255 - grayImage[i,j]
        result[i,j] = np.uint8(gray)

# 显示图像
cv2.imshow("Gray Image", grayImage)
cv2.imshow("Result", result)

# 等待显示
cv2.waitKey(0)
cv2.destroyAllWindows()
```

其输出结果如下图所示，图像处理前后的灰度值是互补的。



图像灰度反色变换在医学图像处理中有一定的应用，如下图所示：



希望文章对大家有所帮助，如果有错误或不足之处，还请海涵。最近连续奔波考博，经历的事情太多，有喜有悲，需要改变自己好好对女神，也希望读者与我一起加油。
(By: Eastmount 2019-03-28 深夜12点 <https://blog.csdn.net/Eastmount/>)