第五章、第六章 练习题。所有练习题都书写交上。对编程题，写出直接相关的函数就可以，不用书写完整程序。

第五章 Image Processing Techniques

1. Which function splits a multichannel into several single-channel images?

2. Which function merges several single-channel images into a multichannel image?

3. Translate an image 150 pixels in the x direction and 300 pixels in the y direction.

4. Rotate an image named img by 30 degrees with respect to the center of the image with a scale factor of 1.

5. Build a 5 x 5 averaging kernel and apply it to an image using cv2.filter2D().

6. Add 40 to all the pixels in a grayscale image

1. 哪个函数将多通道分割成多个单通道图像?

Cv2.split()

1. 哪个函数将多个单通道图像合并成一个多通道图像?

cv2.merge()

1. 在x方向上平移150像素，在y方向上平移300像素。

height, width = image.shape[:2]

M = np.float32([[1, 0, 150], [0, 1, 300]])

dst\_image = cv2.warpAffine(image, M, (width, height))

1. 将名为img的图像相对于图像中心旋转30度，比例系数为1。

height, width = image.shape[:2]

M = cv2.getRotationMatrix2D((width / 2.0, height / 2.0), 30, 1)

dst\_image = cv2.warpAffine(image, M, (width, height))

1. 构建一个5 x 5的平均内核，并使用cv2.filter2D()将其应用于图像。

kernel\_averaging\_5\_5 = np.array([[0.04, 0.04, 0.04, 0.04, 0.04],

[0.04, 0.04, 0.04, 0.04, 0.04],

[0.04, 0.04, 0.04, 0.04, 0.04],

[0.04, 0.04, 0.04, 0.04, 0.04],

[0.04, 0.04, 0.04, 0.04, 0.04]])

smooth\_image\_f2D = cv2.filter2D(image, -1, kernel\_averaging\_5\_5)

6. 将灰度图像中的所有像素加40。

M = np.ones(image.shape, dtype="uint8") \* 40

added\_image = cv2.add(image, M)

第六章 Constructing and Building Histograms

1. What is an image histogram?

2. Calculate the histogram of a grayscale image using 64 bins.

3. Add 50 to every pixel on a grayscale image (the result will look lighter) and calculate the histogram.

4. Calculate the red channel histogram of a BGR image without a mask.

5. What functions do OpenCV, NumPy, and Matplotlib provide for calculating histograms?

6. Modify the grayscale\_histogram.py script to compute the brightness of these three images (gray\_image, added\_image, and subtracted\_image). Rename the script to grayscale\_histogram\_brightness.py.

7. Modify the comparing\_hist\_equalization\_clahe.py script to show the execution time of both cv2.equalizeHist() and CLAHE. Rename it to comparing\_hist\_equalization\_clahe\_time.py.

1. 什么是图像直方图?

图像直方图是一种反映图像色调分布的直方图，描述每个色调值的像素数量。每个色调值的像素数也称为频率。

1. 用64 bins计算灰度图像的直方图。

image = cv2.imread('lenna.png')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

hist = cv2.calcHist([gray\_image], [0], None, [64], [0, 256])

1. 将灰度图像上的每个像素加50，图像更亮，计算直方图。

M = np.ones(gray\_image.shape, dtype="uint8") \* 50

added\_image = cv2.add(gray\_image, M)

hist\_added\_image = cv2.calcHist([added\_image], [0], None, [256], [0, 256])

1. 计算没有掩码的BGR图像的红色通道直方图。

cv2.calcHist([img],[2], None, [256], [0, 256])

1. OpenCV、NumPy和Matplotlib提供了什么函数来计算直方图?

OpenCV提供了计算直方图的cv2.calcHist()函数

NumPy提供了计算直方图的np.histogram()函数

Matplotlib提供了计算直方图的plt.hist()函数

1. 修改grayscale\_histogram.py，计算这三个图像(gray\_image、added\_image和subtracted\_image)的亮度。将脚本重命名为grayscale\_histogram\_brightness.py。()

xg=imread()

[h,s,l]=rgb2hsl(xg);

avg=mean(mean(l));

xa=imread()

[h,s,l]=rgb2hsl(xa);

avg=mean(mean(l));

xs=imread()

[h,s,l]=rgb2hsl(xs);

avg=mean(mean(l));

7.修改comparing\_hist\_equalization\_clahe.py脚本，显示cv2.equalizeHist()和CLAHE的执行时间。将其重命名为comparing\_hist\_equalization\_clahe\_time.py。

start = timer()

image = cv2.imread('lenna.png')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

gray\_image\_eq = cv2.equalizeHist(gray\_image)

end = timer()

exec\_time\_calc\_hist = (end - start) \* 1000

start = timer()

clahe = cv2.createCLAHE(clipLimit=2.0)

gray\_image\_clahe = clahe.apply(gray\_image)

end = timer()

exec\_time\_np\_hist = (end - start) \* 1000