18 机器人 2 班 林宇轩 201810134038

1. 哪个函数将多通道分割成多个单通道图像?

cv2.split()

2. 哪个函数将多个单通道图像合并成一个多通道图像?

cv2.merge()

3. 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)) }
```

4. 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 = img.shape[:2]
M = cv2.getRotationMatrix2D((width / 2.0, height / 2.0), 30, 1)
dst_img = cv2.warpAffine(img, M, (width, height))
```

5. 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],
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(gray_image.shape, dtype="uint8") * 40
added_image = cv2.add(gray_image, M)
```

---

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。

2. 用 64 bins 计算灰度图像的直方图。

```
cv2.calcHist([gray_image], [0], None, [64], [0, 256])
```

3. 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],
```

4.计算没有掩码的 BGR 图像的红色通道直方图。

```
image = cv2.imread('xx.png')
red_image = cv2.cvtColor(image, cv2.COLOR_BGR2RED)
hist = cv2.calcHist([red_image], [0], None, [256], [0, 256])
```

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

```
cv2.calcHist()
np.histogram()
plt.plot()
```

6.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。()

```
xgray=imread()
[h,s,l]=rgb2hsl(xgray)
avg=mean(mean(l))
xadded=imread()
[h,s,l]=rgb2hsl(xadded)
avg=mean(mean(l))
xsubtracted=imread()
[h,s,l]=rgb2hsl(xsubtracted)
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
```