

# 计算机视觉第二次实验

1611640305\_温吉祥

## 实验内容：图像增强

1. 直方图均衡化
2. 对数变换

### 1. 直方图均衡化

1. 实现原理：利用图像直方图对对比度进行调整的方法，使得变换后的图像直方图均匀分布，增加了图像的全局对比度。

2. 主要步骤及核心代码

- 统计灰度直方图

```
img = Image.open('./mountain.jpg')
img_arr = np.array(img)
#统计图片中各灰度值出现的频数，存储在img_hist中
img_hist, bins = np.histogram(img_arr.flatten(), range(256))
```

- 求出每个灰度值的累计和

```
#利用np.cumsum()方法计算累积和
cdf = img_hist.cumsum() #len(cdf) = 255
```

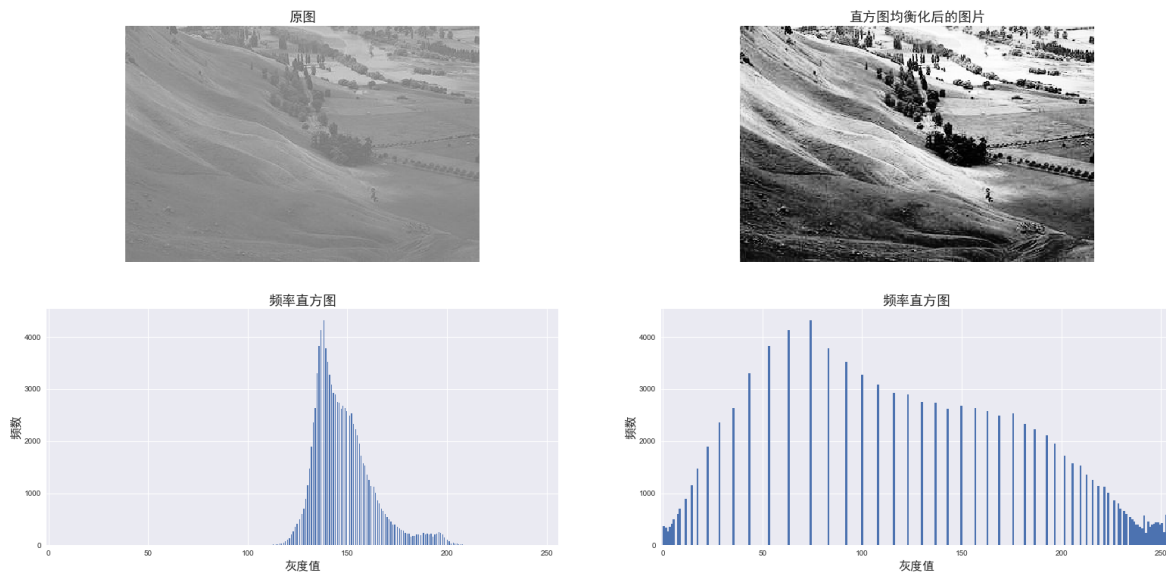
- 求出每个灰度级对应的输出灰度级

```
#获得灰度值映射表, 255* (当前灰度值累计和/灰度值总个数)
map_list = []
for val in cdf:
    map_list.append(int(255*val/cdf[-1]))
```

- 映射到新的图像

```
#根据映射表更新图像灰度值
width,height = img_arr.shape
for i in range(width):
    for j in range(height):
        img_arr[i,j] = map_list[img_arr[i,j]]
```

- 结果对比图



## 2. 对数变换

1. 实现原理：对数变换可以将图像的低灰度值部分扩展，显示出低灰度部分更多的细节；将其高灰度值部分压缩，减少高灰度值部分的细节，从而达到强调图像低灰度部分的目的。

2. 主要步骤及核心代码

- 分离三个通道

```
img = Image.open('./city.jpeg')
img_arr = np.array(img)
rows,cols,_ = img_arr.shape    #(413, 649, 3)
c = 255/np.log(256)           # s = c*log(1+r)
r,g,b = img.split()           #分离三通道
```

- 对数变换函数

```
#对图像做对数变换
def log_transformation(img_arr):
    for i in range(rows):
        for j in range(cols):
            img_arr[i][j] = c * np.log2(1 + img_arr[i][j])
    return Image.fromarray(np.array(img_arr))
```

- 将三个通道分别做对数变换后再合并成彩图

#分别获得三个通道经对数变换的结果

```
img_r = log_transformation(np.array(r))
```

```
img_g = log_transformation(np.array(g))
```

```
img_b = log_transformation(np.array(b))
```

#合并三通道为彩图

```
new_img = Image.merge('RGB', (img_r, img_g, img_b))
```

- 结果对比图

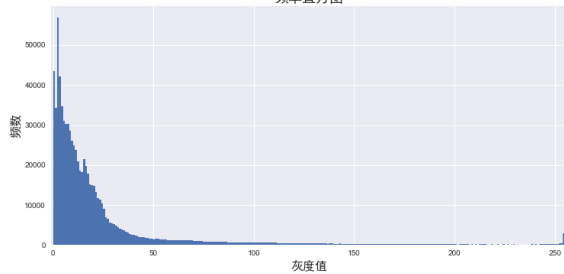
原图



对数变换后的图片



频率直方图



频率直方图

