

pillike

Solarize

将高于阈值的所有像素值反转

1. 接口

```
Solarize(img, n)
```

参数:

- img: 输入图像
- n: 阈值, 高于阈值的所有像素值将被反转

2. 公式原理

设某一点像素值为[r g b], 设n为阈值, 若 $r/g/b \geq n$, 则 $r/g/b = \sim r/g/b$

3. 示例



Posterize

保留Image各通道像素点数值的高bits位, 对应的低(8-bits)个bit置0。

1. 接口

```
Posterize(img, n)
```

参数:

- img: 输入图像
- n: 需要保留的位数, 像素点数值的高n位被保留

2. 公式原理

设某一点像素值为[r g b], 将r、g、b对应的低(8-bits)个bit置0, 即该像素RGB值变为:

$[r \gg (8-bits) \ll (8-bits) \quad g \gg (8-bits) \ll (8-bits) \quad b \gg (8-bits) \ll (8-bits)]$

3. 示例



Equalize/GrayscaleEqualize

图像的直方图均衡

1. 接口

```
GrayscaleEqualize(img) #灰度图像的直方图均衡
Equalize(img) # 彩色图像的直方图均衡
```

参数:

- o img: 输入图像

2. 公式原理

图像直方图: 一个灰度级在 $[0, L-1]$ 的数字图像, 其直方图是一个离散函数

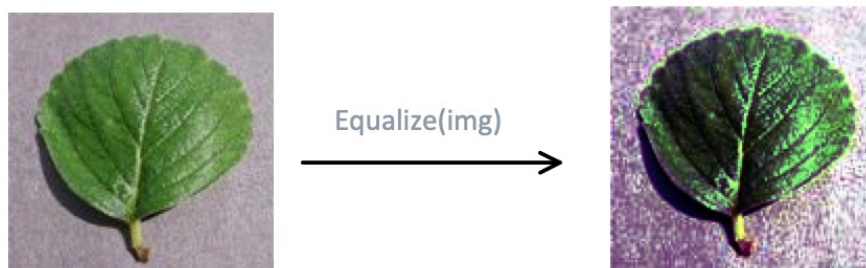
$p(r_k) = n_k/n$, n 是图像的像素总数, n_k 是图像中第 k 个灰度级的像素总数, r_k 是第 k 个灰度级, $k = 0, 1, 2, \dots, L-1$



图 3.18 (a) 一个任意的 PDF; (b) 对所有灰度级 r 应用式(3.3-4)中的变换的结果。具有均匀 PDF 的结果灰度 s 与 r 的 PDF 的形式无关

离散形式的直方图均衡变换函数: $s_k = T(r_k) = (L-1) \sum_{j=0}^k \frac{n_j}{n}$

3. 示例





EnhanceColor

随机调整饱和度

1. 接口

```
EnhanceColor(img)
```

参数：

- img: 输入图像

2. 公式原理

$$L = \frac{1}{2}(max + min)$$

$$S = \begin{cases} \frac{max - min}{max + min} & \text{if } L \leq \frac{1}{2} \\ \frac{max - min}{2 - (max + min)} & \text{if } L > \frac{1}{2} \end{cases}$$

其中，L表示亮度，S表示饱和度，min和max代表RGB空间中的R、G、B颜色值中的最小最大值，范围均为0-1的实数。

- 当增量*i* (increment) ≥ 0 时，此时饱和度采用指数型增长方式调整。当前饱和度*s*与增量*i*之和可能大于1，超出饱和度上限。则选定当*i*+*s*大于1时，取*s*补数作为调整依据（*s*补数越大，调整越大）；当*i*+*s*小于1，取*i*作为调整依据（*i*越大，调整越大）。RGB空间的调整系数alpha公式：

$$\alpha = \begin{cases} \frac{1-s}{s} & \text{if } i+s > 1 \\ \frac{i}{1-i} & \text{if } i+s < 1 \end{cases}$$

其中，*s*代表当前饱和度，*i*代表设置的饱和度增量

$$RGB' = RGB + (RGB - L \times 255) \times \alpha$$

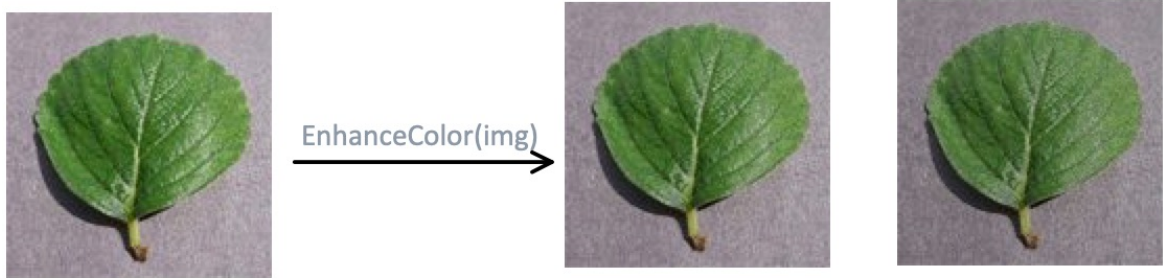
- 当增量*i* (increment) < 0 时，此时饱和度采用线性减弱方式调整。考虑到当*i*取-1时，R、G、B值会相等，可以直接采用线性的方式作调整。RGB空间的调整系数alpha公式：

$$\alpha = i$$

其中，*i*代表设置的饱和度增量

$$RGB' = L \times 255 + (RGB - L \times 255) \times (1 + \alpha)$$

3. 示例



EnhanceContrast

随机调整图像的对比度

1. 接口

```
EnhanceContrast(img)
```

参数:

- img: 输入图像

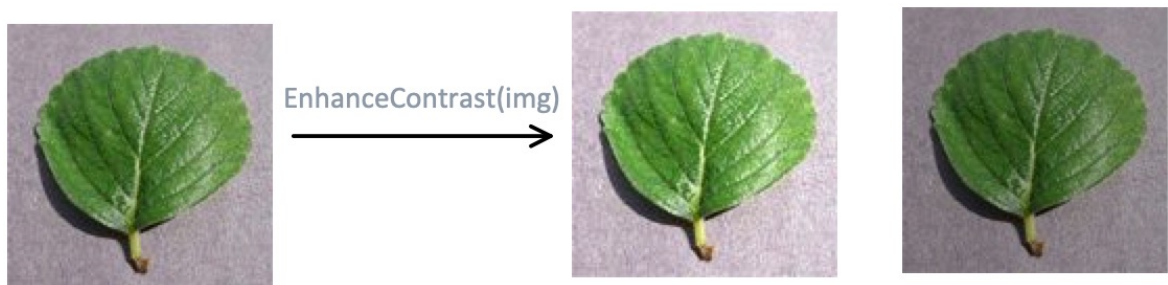
2. 公式原理

对比度、亮度变换:

$$O(r, c) = a * I(r, c) + b, \quad 0 \leq r < H, 0 \leq c < W$$

其中, 输入图像为I, 宽W、高为H, 输出图像为O。a的改变影响图像的**对比度**, b的改变影响图像的**亮度**

3. 示例



EnhanceBrightness

随机调整图像的亮度

1. 接口

```
EnhanceBrightness(img) #非线性调整图像亮度
```

参数:

- img: 输入图像

2. 公式原理

$$O(r, c) = a * I(r, c) + b, \quad 0 \leq r < H, 0 \leq c < W$$

其中，输入图像为I，宽W、高为H，输出图像为O。a的改变影响图像的对比度，b的改变影响图像的亮度

3. 示例



EnhanceSharpness

随机调整图像的锐度，突出、加强图像中景物的边缘和轮廓

1. 接口

```
EnhanceSharpness(img)
```

参数:

- img: 输入图像

2. 公式原理

二阶导数可以通过快速的图像像素值强度的变化来检测图像边缘。Laplacian算子是一种无方向性的二阶导数算子，利用算子，可以进行图像锐化和图像边缘检测，其在点(x+y)处的拉普拉斯值定义为:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(i+1, j) - 2f(i, j) + f(i-1, j)$$

其中,

$$\frac{\partial^2 f}{\partial y^2} = f(i, j+1) - 2f(i, j) + f(i, j-1)$$

常用的Laplacian高通滤波算子有:

$$H_1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & c & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

当c大于8的时候表示图像锐化，越接近8表示锐化效果越好。

当c等于8的时候表示对图像进行高通滤波。

当c值越大的时候，图像锐化效果在减弱，中心像素值的作用在提升。

//另有：USM锐化增强算法

3. 示例

