

# 图像质量衡量指标

最近在做图像去模糊、超分辨率等工作，其中两个重要的定量指标便是 $PSNR$ 和 $SSIM$

## PSNR

- $PSNR$ 的全称是Peak Signal-to-Noise Ration，中文直译为峰值信噪比
- $PSNR$ 是最普遍和使用最为广泛的一种图像客观评价指标，它是基于对应像素点之间的误差，可能会与人眼评价结果不一致的情况
- 计算公式如下：

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (X(i, j) - Y(i, j))^2$$
$$PSNR = 10 \log_{10} \left( \frac{MAX_X^2}{MSE} \right) = 20 \log_{10} \left( \frac{MAX_X}{\sqrt{MSE}} \right)$$

其中， $MSE$ 表示当前图像 $X$ 和参考图像 $Y$ 的均方误差， $H$ 和 $W$ 分别是图像的高度和宽度， $MAX$ 是图像的灰度级，为255

- $PSNR$ 单位是 $dB$ ，数值越大表示失真越小

## SSIM

- $SSIM$ (Structural Similarity)结构相似性，也是一种图像质量评价指标，它分别从亮度、对比度、结构三方面度量图像相似性。
- 计算公式如下：

$$SSIM(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma$$

$$\alpha > 0, \beta > 0, \gamma > 0,$$

其中：

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$
$$c(x, y) = \frac{\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$
$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

- $l(x, y)$ 是亮度比较， $c(x, y)$ 是对比度比较， $s(x, y)$ 是结构比较， $\mu_x$ 和 $\mu_y$ 分别代表 $x, y$ 的平均值， $\sigma_x$ 和 $\sigma_y$ 分别代表 $x, y$ 的标准差， $\sigma_{xy}$ 代表 $x, y$ 的协方差， $c_1, c_2, c_3$ 分别为常数，避免分母为0带来系统错误。

- 在实际操作中，一般设定 $\alpha = \beta = \gamma = 1$ ，以及 $c_3 = c_2/2$ ，可以将 $SSIM$ 简化为以下：

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

- $SSIM$ 具有对称性，即 $SSIM(x, y) = SSIM(y, x)$
- $SSIM$ 是一个0到1之间的数，越大表示输出图像和无失真图像之间的差距越小，即图像质量越好

## 代码

*TensorFlow*框架里有直接计算这两个指标的函数，直接调用就可以了：

```
import tensorflow as tf

def read_img(path):
    return tf.image.decode_png(tf.read_file(path))

def compute_psnr(img1, img2):
    return tf.image.psnr(img1, img2, max_val=255)

def compute_ssim(img1, img2):
    return tf.image.ssim(img1, img2, max_val=255)

def main():
    img1 = read_img('./001.png')
    img2 = read_img('./002.png')

    with tf.Session() as sess:
        sess.run(tf.global_variables_initializer())
        psnr = compute_psnr(img1, img2)
        ssim = compute_ssim(img1, img2)

        print('PSNR = ', sess.run(psnr))
        print('SSIM = ', sess.run(ssim))

if __name__ == '__main__':
    main()
```