




# lsb随机隐写 c语言,lsb.py · Waldeinsamkeit/LSB随机隐写与分析 - Gitee.com

转载

[糖份不足待充值](#)  于 2021-05-24 04:03:33 发布  30  收藏

文章标签: [lsb随机隐写 c语言](#)

# encoding = utf-8

```
import cv2

import random

def dec_to_bin(dec):

    return '{:08b}'.format(dec)

def bin_to_dec(binary_code):

    dec = 0

    for i in range(len(binary_code) - 1):

        dec = dec + int(binary_code[i]) * int(pow(2, 7 - i))

    return dec

# 文件信息转二进制流

def read_data_file(path):

    fp = open(path, "rb")

    stream = ""

    s = fp.read()

    for i in range(len(s)):

        tmp = bin(s[i]).zfill(8)

        stream = stream + tmp.replace('0b', "")

    fp.close()

    return stream

def lsb(image, data_stream, random_index):

    for i in range(len(data_stream)):

        x = random_index[i] % image.shape[0]

        y = int(random_index[i] / image.shape[0])

        value = image[x, y]

        if value % 2 != data_stream[i]:
```

```
if value % 2 == 1:
    image[x, y] = value - 1
else:
    image[x, y] = value + 1
return image

if __name__ == "__main__":
    text_path = './data/data.txt'
    img_path = './image/original.png'
    out_path = './image/steg.png'
    stream = read_data_file(text_path)
    img = cv2.imread(img_path, 0)
    pixel_len = img.shape[0] * img.shape[1]
    rate = len(stream) / pixel_len
    print('隐写率: ', rate)
    if rate <= 1:
        random_ls = random.sample(range(0, pixel_len), len(stream))
        random_ls.sort()
        new_img = lsb(img, stream, random_ls)
        cv2.imwrite(out_path, new_img)
        print('success')
    else:
        print('数据过大')
```

[一键复制](#)

[编辑](#)

[Web IDE](#)

[原始数据](#)

[按行查看](#)

[历史](#)