计算机视觉与模式识别作业四（**2021** 版本）

姓名：	刘沛鑫	学号：	2195011999
班级：	计算机 93	得分：	

1、 给定一幅分辨率为 9×9，图像深度为 8bit 的灰度图，同时给定线性几何变换的参数 A 和 b，变换后的几何位置$[x', y']^T$和变换前的像素位置$[x, y]^T$之间存在变换关系如下所示：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + b$$

1	10	19	28	37	46	55	64	73
2	11	20	29	38	47	56	65	74
3	12	21	30	39	48	57	66	75
4	13	22	31	40	49	58	67	76
5	14	23	32	41	50	59	68	77
6	15	24	33	42	51	60	69	78
7	16	25	34	43	52	61	70	79
8	17	26	35	44	53	62	71	80
9	18	27	36	45	54	63	72	81

尝试：

（1） 、假设图像现在围绕着它的中心点逆时针方向旋转 90°，试问 A 和 b 分别是多少？

（2） 、试问旋转后的图像是什么？

（3） 、写出后向变换的伪代码，首先确定变换后的区域，然后遍历输出图像的像素点位置，计算它在输入图像中的对应位置。

（1）设图像左上角为坐标原点$(0,0)$，向下和向右分别为 x 轴和 y 轴，则图像的中心点坐标为$(4,4)$。绕中心点逆时针方向旋转 90°，在此坐标系中对应旋转角度 $\boldsymbol{\theta = 90°}$

所以

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \end{bmatrix} \right) + \begin{bmatrix} 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 8 \\ 0 \end{bmatrix}$$

由此得到

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 8 \\ 0 \end{bmatrix}$$

（2）旋转后的图像的每个像素用矩阵表示如下：

$$\begin{bmatrix} 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 & 81 \\ 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 \\ 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 \\ 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 \\ 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 \\ 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 \\ 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

（3）假设原图像的区域是$w \times h$（即 9×9）

变换后的区域是$new_w \times new_h$（在此例中也为 9×9），且

$$new\_w = \lceil |w \times cos\theta + h \times sin\theta| \rceil$$

$$new\_h = \lceil |h \times cos\theta + w \times sin\theta| \rceil$$

逆变换

$$\begin{bmatrix} x \\ y \end{bmatrix} = A^{-1} \left(\begin{bmatrix} x' \\ y' \end{bmatrix} - b \right) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} 0 \\ 8 \end{bmatrix}$$

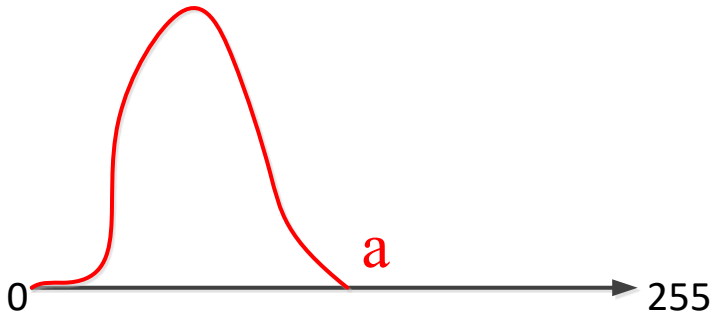
等价于

$$\begin{cases} x = y' \\ y = 8 - x' \end{cases}$$

后向变换代码：

```
new_w = math.ceil(abs(w*cosTHETA + h*sinTHETA))
new_h = math.ceil(abs(h*cosTHETA + w*sinTHETA))
for i in range(new_w):
  for j in range(new_h):
    x = j    #此处根据旋转角度不同而不同
    y = 8 – i     #此处根据旋转角度不同而不同
    x = int(x)
    y = int(y)
    if x >= 0 and x < w and y >= 0 and y < h:
      Out_img[i,j] = In_img[x,y]
```

2、 如果我们现在拍摄了一幅图像，它的直方图如下所示：



尝试：

（1） 、描述这幅图像的缺陷；

（2） 、如果采用一个灰度变换去校正这幅图像，灰度变换函数应该如何设计，试举例说明，并且描述该函数的扩展/压缩特性。

（3） 、如果采用直方图均衡来校正这幅照片，尝试通过伪代码描述矫正的过程。

（1）这幅图像的像素值集中分布在区间[0, a]，未能有效利用量化范围，图像总体偏暗。

（2）由于这幅图像的像素值集中分布在低灰度区，需要扩展低灰度区，可以使用线性变换来实现。

$$t = \begin{cases} \dfrac{255}{a}s, & s \le a \\ 255, & s > a \end{cases}$$

该函数将低灰度区[0, a]扩展到[0, 255]，将高灰度区[a, 255]压缩到[255, 255]。

（3）假设原图像的大小是$w \times h$

```
#统计 1-256 灰度
s = numpy.zeros(256)
for i in range(w):
   for j in range(h):
      s[In_img[i,j]] += 1
#统计概率密度
for i in range(256):
   p[i] = s[i]/(h*w*1.0)
#统计累计概率分布
f[0] = p[0]
```

```
for i in range(1,256):
    f[i] = f[i-1]+ p[i]
#对灰度值进行映射，均衡化
for i in range(w):
    for j in range(h):
        Out_img[i,j] = round(255*f[In_img[i,j]])
```

3、如果我们采用查找表的方式来实现如下的伽马变换：

$t = s^2$

试问：下表中变换后的值应该是多少？

1	20	30	40	50	60	70	80	90	100
0	**2**	**4**	**6**	**10**	**14**	**19**	**25**	**32**	**39**

假设像素深度是 8bit，用 T 和 S 表示像素值，分别归一化为 t 和 s，则

$$t = \frac{T}{255}$$

$$s = \frac{S}{255}$$

由 $t = s^2$ 可得

$$T = \frac{S^2}{255}$$

变换结果如表中所示。

4、给定一幅分辨率为 9×9，图像深度为 3bit 的灰度图，：

	1	2	3	4	5	6	7	8	9
1	6	7	6	3	5	3	0	5	6
2	7	1	7	5	2	3	3	7	1
3	1	7	5	1	7	5	7	7	7
4	7	7	0	5	0	5	2	4	2
5	5	3	6	0	3	6	4	1	1
6	0	6	7	2	3	2	1	1	2
7	2	1	5	0	6	5	6	2	4
8	4	3	6	0	6	5	2	6	3
9	7	7	5	6	1	1	4	2	2

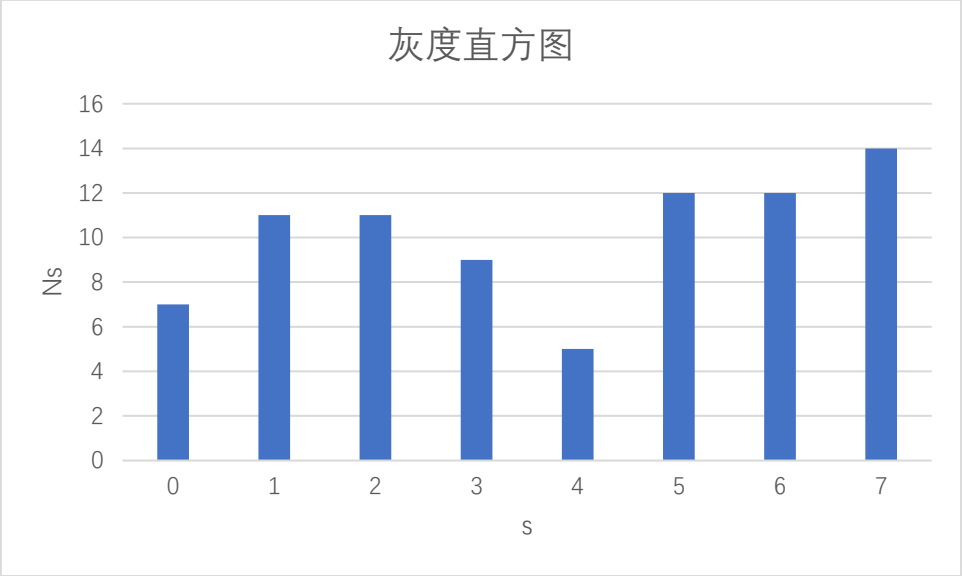
（1） 、计算这幅图像的直方图

（2） 、对这副图像进行直方图均衡

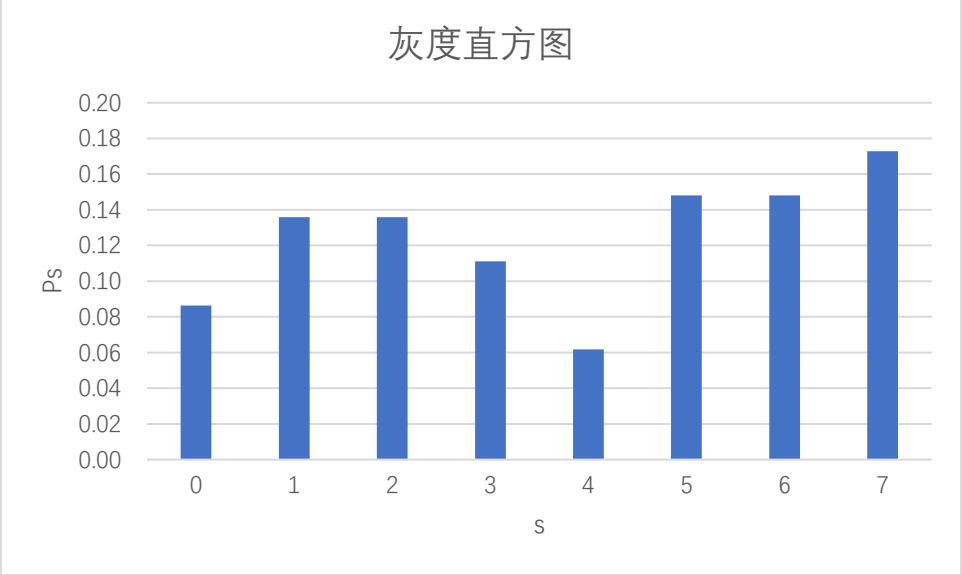
（3） 、尝试分析为什么直方图均衡能够实现图像增强的目的。

（1）直方图统计如下

$$\boldsymbol{N = [7, 11, 11, 9, 5, 12, 12, 14]}$$



$$\boldsymbol{P = [0.0864, 0.1358, 0.1358, 0.1111, 0.0617, 0.1481, 0.1481, 0.1728]}$$

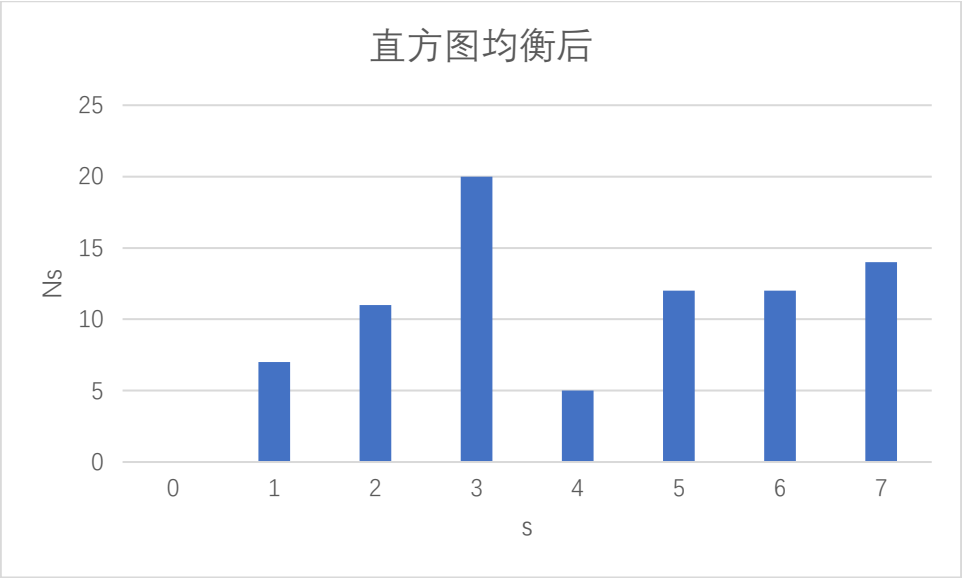


（2）累计概率分布 $\boldsymbol{F_s = \sum_{i=0}^{s} P_i}$

$$\boldsymbol{F = [0.0864, 0.2222, 0.358, 0.4691, 0.5309, 0.679, 0.8272, 1]}$$

灰度变换 $\boldsymbol{T_s = Round(7 \times F_s)}$

T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
1	2	3	3	4	5	6	7



（3）因为直方图均衡化处理之后，原来比较少像素的灰度会被分配到别的灰度去，像素相对集中，处理后对比度变大，清晰度变大，所以能有效增强图像。

5、给定一个 100×100 的图像，在不申请额外存储空间的情况下，尝试对该图像进行顺时针方向的 90° 旋转，并且给出伪代码。

在不申请额外存储空间的情况下，可以对 100×100 图像的 50 层逐层旋转，直接在原始图像上进行修改。

```
for layer in range(50):
    start = layer
    end = 99-layer
    for i in range(end-start):
        temp = img[start][start+i]
        img[start][start+i] = img[end-i][start]
        img[end-i][start] = img[end][end-i]
        img[end][end-i] = img[start+i][end]
        img[start+i][end] = temp
```