FAN Shun 57445835

Problem 1

a) if $D_A = 255$, $D_B = 255 = k \cdot 255 \cdot 255$

$\Rightarrow k = \frac{1}{255}$

b) $D_B = \frac{1}{255} D_A^2$

$f^{-1}(D_B) = \sqrt{255 D_B}$

$$H_B(D_B) = \frac{H_A(f^{-1}(D_B))}{f'(f^{-1}(D_B))} = \frac{H_A(\sqrt{255 D_B})}{\frac{1}{2} \cdot \sqrt{255} (D_B)^{-\frac{1}{2}}} = \frac{2\sqrt{D_B}\, H_A(\sqrt{255 D_B})}{\sqrt{255}}$$

c) $$H_B(D_B) = \frac{2\sqrt{D_B}\, H_A(\sqrt{255 D_B})}{\sqrt{255}} = \frac{2\sqrt{D_B} \cdot \frac{1}{255}}{\sqrt{255}} = \frac{2\sqrt{D_B}}{(255)^{\frac{3}{2}}}$$

Problem 2:

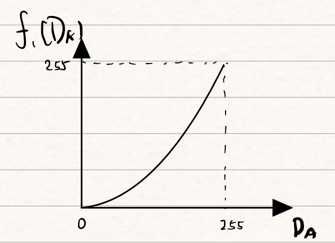
a) $$f_1(D_A) = \frac{D_m}{A_0} \int_0^{D_A} H_A(u)\,du$$

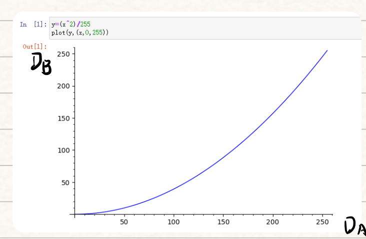
$$= \frac{D_m}{A_0} \int_0^{D_A} \frac{2A_0 u}{D_M^2}\,du$$

$$= \frac{D_m}{A_0} \cdot \frac{2A_0}{D_m^2} \cdot \int_0^{D_A} u\,du$$

$$= \frac{2}{D_m} \cdot \frac{D_A^2}{2}$$

$$= \frac{D_A^2}{D_m} = \frac{D_A^2}{255}$$





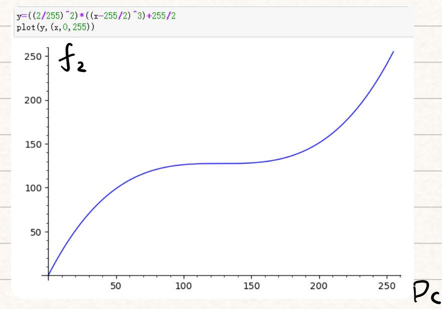
b) $\int_0^{D_m} k\left(u-\frac{255}{2}\right)^2 du$

$$= k\cdot \int_0^{255}\left(u-\frac{255}{2}\right)^2 du = k\cdot\left(\frac{1}{3}\left(\frac{255}{2}\right)^3+\frac{1}{3}\left(\frac{255}{2}\right)^3\right)$$

$$= k\cdot \frac{2}{3}\left(\frac{255}{2}\right)^3$$

$$= A_0$$

$\Rightarrow \quad k = \frac{3}{2}\left(\frac{2}{255}\right)^3\cdot A_0$



$$f_2(D_c) = \frac{D_m}{A_0}\int_0^{D_c} H_c(u)\,du$$

$$= \frac{D_m}{A_0}\int_0^{D_c} k\left(u-\frac{255}{2}\right)^2 du$$

$$= \frac{3}{2}\cdot\frac{255}{A_0}\cdot\left(\frac{2}{255}\right)^3\cdot A_0\cdot\int_0^{D_c}\left(u-\frac{255}{2}\right)^2 du$$

$$= 3\cdot\left(\frac{2}{255}\right)^2\cdot\left[\frac{1}{3}\cdot\left(D_c-\frac{255}{2}\right)^3+\frac{1}{3}\left(\frac{255}{2}\right)^3\right]$$

$$= \left(\frac{2}{255}\right)^2\cdot\left(D_c-\frac{255}{2}\right)^3+\frac{255}{2}$$

c) $f_1(D_A) = \frac{D_A^2}{255}$ $\qquad f_2(D_c) = \left(\frac{2}{255}\right)^2\cdot\left(D_c-\frac{255}{2}\right)^3+\frac{255}{2}$

$D_B = f_1(D_A)$

$D_B = f_2(D_c)$

$$D_c = f_2'(D_B) = \sqrt[3]{\left(D_B-\frac{255}{2}\right)\cdot\left(\frac{255}{2}\right)^2}+\frac{255}{2}$$

$D_c = f_2'(f_1(D_A))$

$$= \sqrt[3]{\left(\frac{D_A^2}{255}-\frac{255}{2}\right)\cdot\left(\frac{255}{2}\right)^2}+\frac{255}{2}$$

Problem 3.

a) $D_B = f(D_A) = \frac{D_m}{A_0} \sum_{0}^{D_A} H_A(i)$

$D_m = 7 \qquad A_0 = 16384$

D_c	$H_c(D_c)$	D_B
0	0	0
1	0	0
2	4096	2
3	4096	4
4	4096	5
5	4096	7
6	0	7
7	0	7

D_B	$D_c = f^{-1}(D_B)$
0	0
1	0
2	2
3	2
4	3
5	4
6	4
7	5

D_A	$H_A(D_A)$	D_B	D_c
0	609	0	0
1	3298	2	2
2	2150	3	2
3	3979	4	3
4	312	4	3
5	1768	5	4
6	426	5	4
7	3842	7	5

$\Rightarrow$

D_A	D_c
0	0
1	2
2	2
3	3
4	3
5	4
6	4
7	5

b) set output as e

D_A	$H_A(D_A)$
0	609
1	3298
2	2150
3	3979
4	312
5	1768
6	426
7	3842

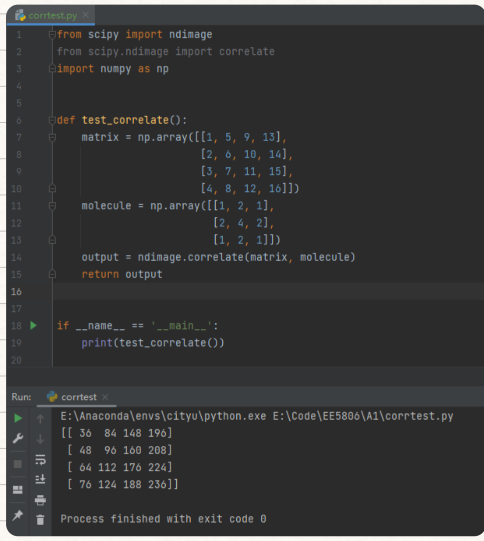
$\Rightarrow$

D_e	$H_e(D_e)$
0	609
1	0
2	5446
3	4291
4	2194
5	3842
6	0
7	0

The output histogram has some characteristics of the target histogram, but it is not a perfect match.
Because the two histograms processed are **discrete**, the point operation will only make the histograms as similar as possible, but the **number of pixels** on each gray level **cannot be changed**.

Problem 4.

a)



```python
from scipy import ndimage
from scipy.ndimage import correlate
import numpy as np


def test_correlate():
    matrix = np.array([[1, 5, 9, 13],
                       [2, 6, 10, 14],
                       [3, 7, 11, 15],
                       [4, 8, 12, 16]])
    molecule = np.array([[1, 2, 1],
                         [2, 4, 2],
                         [1, 2, 1]])
    output = ndimage.correlate(matrix, molecule)
    return output


if __name__ == '__main__':
    print(test_correlate())
```

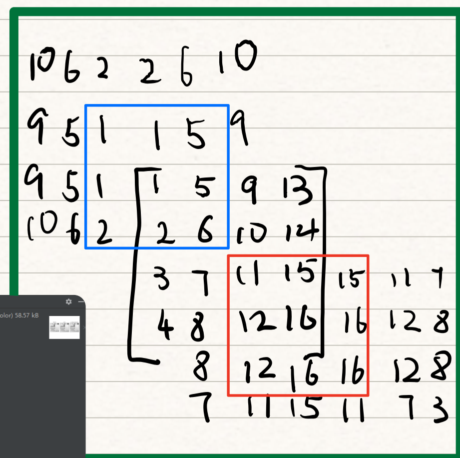
```
E:\Anaconda\envs\cityu\python.exe E:\Code\EE5806\A1\corrtest.py
[[ 36  84 148 196]
 [ 48  96 160 208]
 [ 64 112 176 224]
 [ 76 124 188 236]]

Process finished with exit code 0
```

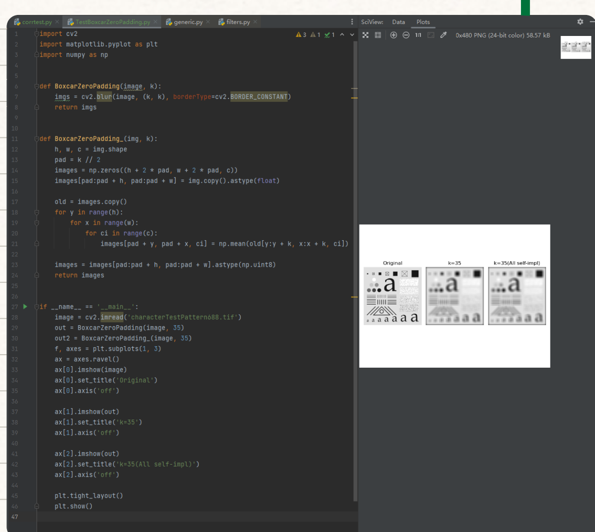
b)

$$I_o(0,0) = 1+1\times2+5+1\times2+1\times4+5\times2+2+2\times2+6$$
$$= 36$$

$$I_o(3,3) = 11+15\times2+15$$
$$+12\times2+16\times4+16\times2$$
$$+12+16\times2+16$$
$$= 236$$



c) d)



```python
import cv2
import matplotlib.pyplot as plt
import numpy as np


def BoxcarZeroPadding(image, k):
    imgs = cv2.blur(image, (k, k), borderType=cv2.BORDER_CONSTANT)
    return imgs


def BoxcarZeroPadding_(img, k):
    h, w, c = img.shape
    pad = k // 2
    images = np.zeros((h + 2 * pad, w + 2 * pad, c))
    images[pad:pad + h, pad:pad + w] = img.copy().astype(float)

    old = images.copy()
    for y in range(h):
        for x in range(w):
            for ci in range(c):
                images[pad + y, pad + x, ci] = np.mean(old[y:y + k, x:x + k, ci])

    images = images[pad:pad + h, pad:pad + w].astype(np.uint8)
    return images


if __name__ == '__main__':
    image = cv2.imread('characterTestPattern688.tif')
    out = BoxcarZeroPadding(image, 35)
    out2 = BoxcarZeroPadding_(image, 35)
    f, axes = plt.subplots(1, 3)
    ax = axes.ravel()
    ax[0].imshow(image)
    ax[0].set_title('Original')
    ax[0].axis('off')

    ax[1].imshow(out)
    ax[1].set_title('k=35')
    ax[1].axis('off')

    ax[2].imshow(out)
    ax[2].set_title('k=35(All self-impl)')
    ax[2].axis('off')

    plt.tight_layout()
    plt.show()
```


Implement detail of c) and d)

e) Images after zero-padding have a significant **black ring around the outer edge** of the image after using a boxcar filter.
This phenomenon is because the values of the images at the edges are averaged to a lower level, i.e., darker after boxcar filtering after the outer edge has been zero-padding.