

影像處理 HW3-2

312512049 電控碩 顏志憲

1. Histogram Statistics method:

```
import cv2
import numpy as np

k0 = 0
k1 = 0.3
k2 = 0
k3 = 0.049

def statistical_measures(img, kernal_size):
    if kernal_size == 0:
        return cv2.meanStdDev(img)[0], cv2.meanStdDev(img)[1]
    else:
        i = kernal_size
        kernal = np.ones((i,i), np.float32)/(i**2)
        mean = cv2.filter2D(img.astype(np.float32), -1, kernal)
        var = cv2.filter2D((img.astype(np.float32)-mean)**2, -1, kernal)
        return mean, np.sqrt(var)

def boundary(a,b):
    a_lower_bound = k0*a
    a_upper_bound = k1*a
    b_lower_bound = k2*b
    b_upper_bound = k3*b
    return [a_lower_bound, a_upper_bound, b_lower_bound, b_upper_bound]

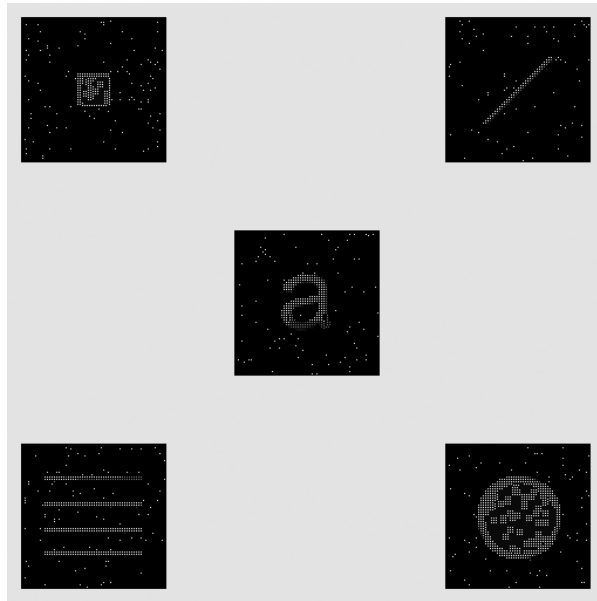
def histogram_statistics(img, local_ms, max, b):
    width, height = img.shape

    for i in range(1, width):
        for j in range(1, height):
            local = img[i-1:i+2, j-1:j+2]
            local_max = np.max(local)
            if (b[0] < local_ms[0][i,j] < b[1]) and (b[2] < local_ms[1][i-1,j-1] < b[3]):
                c = round(max/local_max)
                img[i,j] = round(c*img[i,j])
    return img

if __name__ == '__main__':
    img = cv2.imread('hidden object.jpg', cv2.IMREAD_GRAYSCALE)
    local = statistical_measures(img, 3)
    max = np.max(img)
    mean_std = statistical_measures(img, 0)
    # histogram_statistics method
    b = boundary(mean_std[0], mean_std[1])
    img = histogram_statistics(img, local, max, b)
    cv2.imwrite('histogram_statistics' + '_img.png', img)
```

整個做法如下:首先讀取圖片後,首先利用一個總和為一的 3*3 filter kernel 來對圖像進行平滑處理計算得到影像得局部平均值和標準差以及圖片中像素的最大值,隨後計算原圖的全域像素平均值和標準差,接著利用設定好的係數來計算均值和方差的邊界(係數透過手動調整),最後利用這個邊界、和影像以及影像的局部均值和標準差來增強圖片,具體方法首先提取影像局部,並尋找最大像素值,當該局部影像的均值及方差都在邊界內,則將該區域內的像素乘上一個係數來對影像進行增強,其中該系數計算方式為利用全域最大值除該區域局部最大值。

銳化結果：

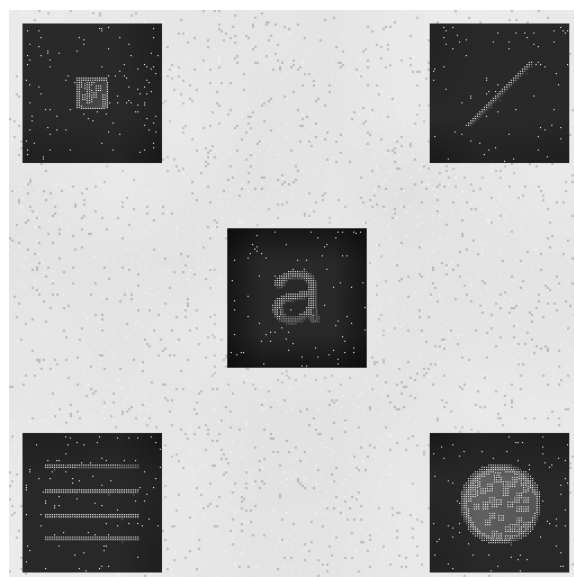


2. Local enhancements method:

使用 OpenCV 內建的函式庫來進行局部影像增強，該函式將影像切割為圖塊，並對每個圖塊進行直方圖均衡化，並限制對比度，以防止過大的雜訊

```
# local enhancement method
clahe = cv2.createCLAHE()
clahe_img = clahe.apply(img)
cv2.imwrite('local enhancement' + '_img.png', clahe_img)
```

銳化結果：



3. Comment:

局部直方圖均衡化是計算每個像素鄰域內的直方圖，並進行直方圖均衡化得到轉換函數，隨後裡用該函數映射鄰域中心像素的強度並重複以上步驟。而直方圖統計方法則是使用全域的像素均值及方差來做局部的增強，以局部的均值及方差來做為是否改變強度的依據。由結果可以發現，直方圖統計法相比局部增強法有更好得增強效果，保留了原本較亮背景的平滑，抑制了許多雜訊。