Hw1_Spatial Image Enhancement

學號:610415145 姓名:許晉偉

Date due:2022/11/08

Date handed in:2022/11/07

■Technical description

(1) power-law (gamma) transformation

```
import cv2
import numpy as np
```

使用 opencv 函式庫進行基本圖片操作

使用 numpy 函式庫進行矩陣建立

```
c=255
y=float(input("y="))
for i in ["Cameraman.bmp",'Peppers.bmp','Jetplane.bmp','Lake.bmp']
    img = cv2.imread(i,0)
    H,W= img.shape[:]
    gamma=np.zeros((H,W),dtype='uint8')
    #s=cr^y
```

一開始依照公式設定參數 $s = cr^{\gamma}$

C=是常數用於將影像數值正規化到[0~1]區間最後在放大回 0~255

所以設定成 255,接者設定指數 y 用於調整明亮度,接者依序用 for 迴圈

讀入圖片並用 gamma 這個變數存放伽瑪校正後的圖

```
    for i in range(H):
        for j in range(W):
            gamma[i][j] = c*((img[i][j]/c)**y)
    result = cv2.hconcat([img,gamma])
    cv2.imshow('result',result)
    cv2.waitKey(0)
cv2.destroyAllWindows()
```

對於原圖套入公式後存入 gamma 陣列裡並將原圖和結果一起顯示出來

(2) histogram equalization

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

img = cv2.imread('Cameraman.bmp',0)
#img = cv2.imread('Peppers.bmp',0)
#img = cv2.imread('Jetplane.bmp',0)
#img = cv2.imread('Lake.bmp',0)
H,W = img.shape[:]
pdf= np.zeros(256)
cdf= np.zeros(256)
```

使用 opencv 函式庫進行基本圖片操作

使用 numpy 函式庫進行矩陣建立

使用 plt 函式庫用於顯示直方圖

建立空陣列存放函數所需的值

```
for i in range(H):
    for j in range(W):
        pdf[img[i][j]]+=1
s=H*W
for i in range(len(pdf)):
    cdf[i]=pdf[i]/s
for i in range(1,256):
    cdf[i]=cdf[i]+cdf[i-1]
```

$p_r(r_k) = \frac{n_k}{n}$ 利用迴圈計算出 PDF 的值

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, \ldots, L-1.$$

使用此公式以 for 迴圈將 pdf 累加至 cdf

陣列

```
img_hist=img.copy()
m=np.min(cdf)
M=np.max(cdf)
for i in range(H):
    for j in range(W):
        img_hist[i][j]=round((cdf[img[i][j]]-m)/(M-m)*255)
```

img_hist 這個變數用來存放代入公式後的結果圖,使

用 copy 函式,是因為物件的特性避免修改到原圖,接 $h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{cdf_{max} - cdf_{min}} \times (L-1)\right)$

者套入直方圖均衡化算式將各像素以本身輸入的像

素分佈機率進行重新分佈，達成將集中的像素分散的成果

```
plt.subplot(211)
plt.hist(img.ravel(),256,[0,255])
plt.subplot(212)
plt.hist(img_hist.ravel(),256,[0,255])
plt.savefig('histogram1.png')
plt.show()
result = cv2.hconcat([img,img_hist])
cv2.imshow("result",result)
cv2.imwrite('his1.png', result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

接者使用 plot 函式顯示原圖和

結果圖灰階分布的直方圖

並用 cv2 函式將原圖和直方均衡

化的結果圖顯示出來

(3) image sharpening by Laplacian

```
import cv2
import numpy as np
img = cv2.imread('Cameraman.bmp',0)
#img = cv2.imread('Peppers.bmp',0)
#img = cv2.imread('Jetplane.bmp',0)
#img = cv2.imread('Lake.bmp',0)
H,W = img.shape[:]
img2= np.zeros((Height, width), np.uint8)
k=np.array([[-1,-1,-1],[-1,9,-1],[-1,-1,-1]])
```

使用 opencv 函式庫進行基本

圖片操作

使用 numpy 函式庫進行矩陣

建立

k 變數為參考上課講義使用簡化後的 kernel

這樣卷積後就可以少掉與原圖相加的步驟

−1	−1	−1
−1	9	−1
−1	−1	−1

img2 變數用於存放結果圖

```
def convolution(x,y,img):
    result=0
    for i in range(0,3):
        for j in range(0,3):
            result=result+img[x+i,y+j]*k[i][j]
    return result

for i in range(0,H-2):
    for j in range(0,W-2):
        temp=convolution(i,j,img)
        if(temp)<0 :
            temp=0
        if(temp)>255 :
            temp=255
        img2[i][j]=temp
```

接者利用定義的卷積函式進行卷積後

進行 Relu 避免卷積後的值超過

(0~255)這個範圍造成結果錯誤

所以我用 temp 這個變數判斷後再存

入結果圖

```
img = cv2.resize(img, (H-2, W-2), interpolation=cv2.INTER_AREA)
result = cv2.hconcat([img,img2])
cv2.imwrite('Laplacian1.png', img2)
cv2.imshow('result',result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

接者利用 cv2 圖函式庫顯示出原圖和銳化過的圖

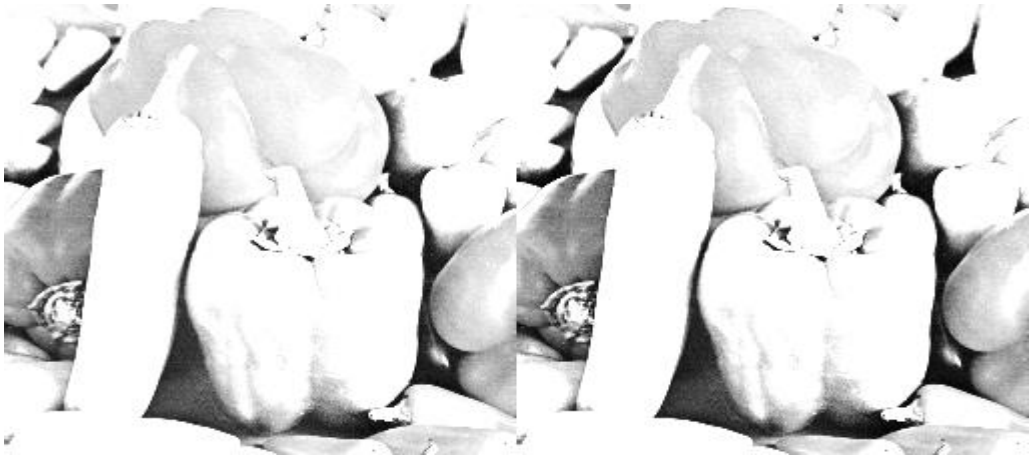
■Experimental results

(1) power-law (gamma) transformation

原圖　　power-law

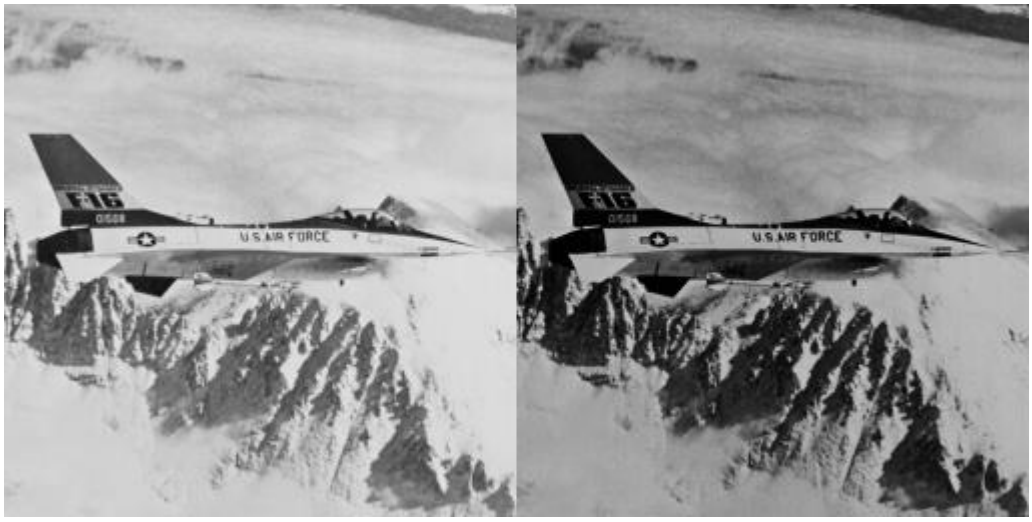
'Cameraman.bmp' y=1.2

'Peppers.bmp' y=1.3



原圖 power-law

'Jetplane.bmp' y=2.2

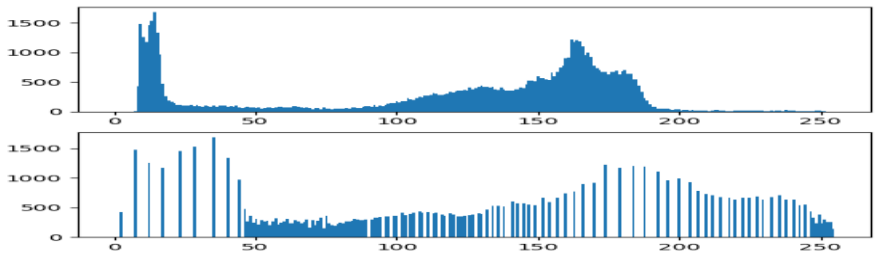


'Lake.bmp' y=0.4

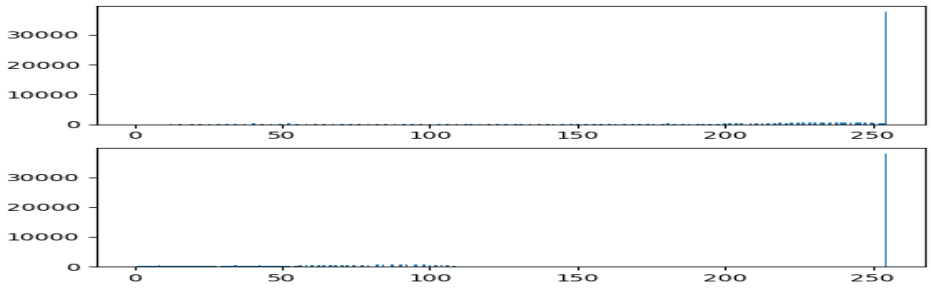
(2) histogram equalization

原圖　　histogram equalization

'Cameraman.bmp'

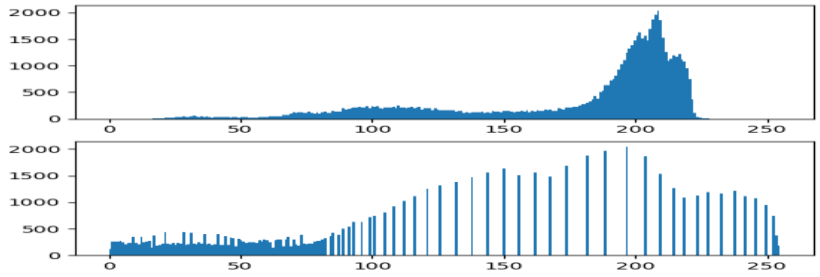


'Peppers.bmp'

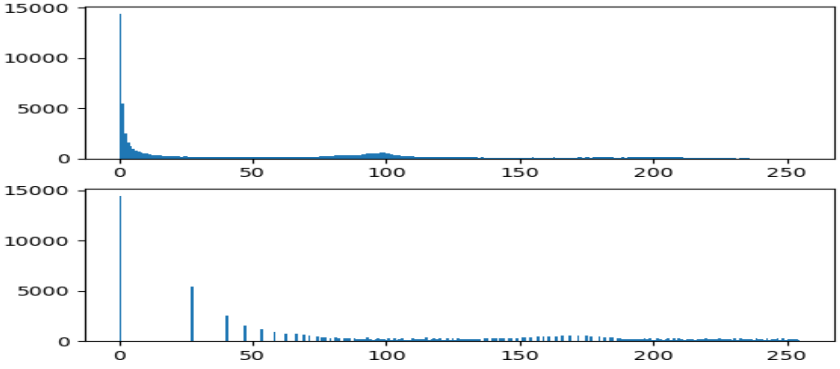


原圖　　histogram equalization

'Jetplane.bmp'



'Lake.bmp'



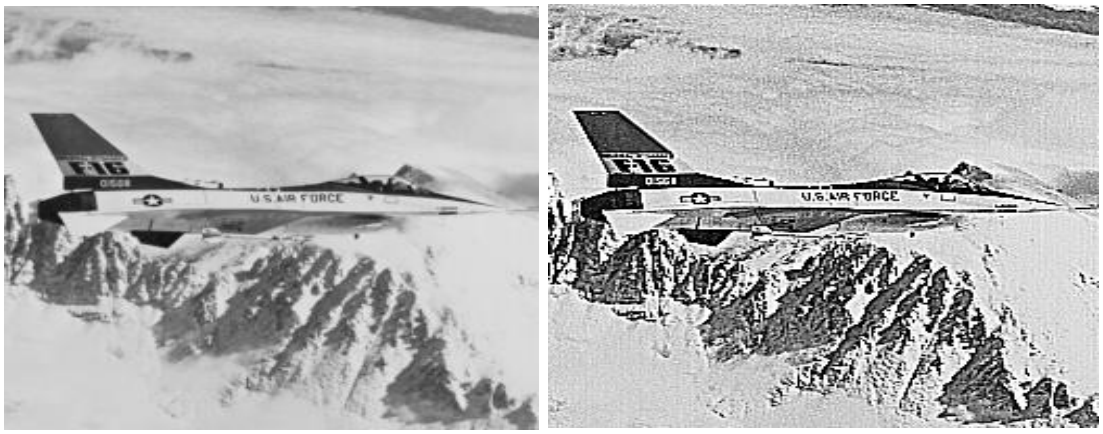
(3) image sharpening by Laplacian

原圖　　image sharpening by Laplacian

'Cameraman.bmp'

'Peppers.bmp'

'Jetplane.bmp'



原圖 image sharpening by Laplacian

'Lake.bmp'


■Discussions

針對三種圖像增強方法

(1) power-law (gamma) transformation

我覺得這是一個非常簡單且有效的增加對比度的方法,對過濾雜訊並不顯著。

(2) histogram equalization

相比前一個方法,直方圖實現的過程複雜很多,但是效果也比伽瑪校正好很多透過重新分配像素照片也很自然的提高對比度。

(3) image sharpening using the Laplacian

一開始我沒做 Relu 圖片的結果呈現的很怪,一讀取圖片的值經過驗算後才發現溢位了,之後補上 Relu 後結果就正常了。

■ References and Appendix

https://blog.xuite.net/viplab/blog/307263602-

mage+Enhancement+in+the+Spatial+Domain

https://iris123321.blogspot.com/2017/05/histogram-equalization.html

https://zh.wikipedia.org/wiki/%E7%9B%B4%E6%96%B9%E5%9B%BE%E5

%9D%87%E8%A1%A1%E5%8C%96

https://ithelp.ithome.com.tw/articles/10192114