program code :

import cv2

import numpy as np

from matplotlib import pyplot as plt

p1 = cv2.imread('picture1.jpg',0)

p2 = cv2.imread('picture2.png',0)

def showPicture(pic,title): # pic : 圖片 ; title : 標題

plt.subplot(111)

plt.imshow(pic, cmap = 'gray')

plt.title(title)

plt.xticks([]), plt.yticks([])

plt.show()

# show original picture

showPicture(p1,'Original Picture 1 (gray scale)')

showPicture(p2,'Original Picture 2 (gray scale)')

# show Gaussian Blurred picture # 模糊化處理

p1\_2 = cv2.GaussianBlur(p1,(9,9),0) # 9×9的Gaussian模糊化

showPicture(p1\_2,'Gaussian Blurred Picture 1 (gray scale)')

p2\_2 = cv2.GaussianBlur(p2,(7,7),0)

showPicture(p2\_2,'Gaussian Blurred Picture 2 (gray scale)')

# show Equalized Histogram picture # 將圖片Histograms圖形分布較平均

p1\_3 = cv2.equalizeHist(p1\_2)

showPicture(p1\_3,'Equalized Histogram Picture 1 (gray scale)')

p2\_3 = cv2.equalizeHist(p2\_2)

showPicture(p2\_3,'Equalized Histogram Picture 2 (gray scale)')

# show Sobel operator filtered picture # Sobel邊緣檢測演算法

sX1 = cv2.Sobel(p1\_3, cv2.CV\_16S, 1, 0, ksize = 3) # cv2.Sobel(src, ddepth, dx, dy, ksize)

sY1 = cv2.Sobel(p1\_3, cv2.CV\_16S, 0, 1, ksize = 3) # ddepth:影象的深度，-1表示與原影象相同 ; dx,dy:求導的階數 ;

# ksize:Sobel運算元的大小，必須為1、3、5、7。

# cv2.CV\_16S : 16位有符號的資料型別 (Sobel函式求完導數後有負值和大於255之值)

sobelX1 = cv2.convertScaleAbs(sX1) # convertScaleAbs():轉回原來的uint8形式

sobelY1 = cv2.convertScaleAbs(sY1)

final\_p1 = cv2.addWeighted(sobelX1, 3, sobelY1, 3, -300) # cv2.addWeighted():組合兩個方向的計算

# cv2.addWeighted(src1, alpha, src2, beta, gamma)

# alpha:權重1 ; beta:權重2 ; gamma:加到結果上的值

showPicture(final\_p1,'Edge Picture 1')

sX2 = cv2.Sobel(p2\_3, cv2.CV\_16S, 1, 0, ksize = 3)

sY2 = cv2.Sobel(p2\_3, cv2.CV\_16S, 0, 1, ksize = 3)

sobelX2 = cv2.convertScaleAbs(sX2)

sobelY2 = cv2.convertScaleAbs(sY2)

final\_p2 = cv2.addWeighted(sobelX2, 6, sobelY2, 9, -500)

showPicture(final\_p2,'Edge Picture 2')

methods :

(1) 顯示原本的圖片，確認讀取。

(2) 將圖片Gaussian模糊化處理。分別做9×9的Gaussian模糊化和7×7的Gaussian模糊化。

(3) 對圖片進行Equalized Histogram的處理。

(4) 用Sobel邊緣檢測演算法，輸出邊緣輪廓的圖片。

Bonus

Other ideas for improving the detection result :

(1) 將cv2.Sobel(src, ddepth, dx, dy, ksize)的ddepth從-1改成cv2.CV\_16S。

(2) 在cv2.addWeighted(src1, alpha, src2, beta, gamma)中，將權重alpha和beta設為大於1，依要強化的不同方向圖的邊緣而調整，但也會強化到雜訊，再令gamma為負值調整所有結果。最後突顯了主要的邊緣輪廓。

results :















