

影像處理概論 HW4

題目：Color edge detection

409410095 王子奕

繳交期限：June 16, 2023

繳交日期：June 15, 2023

● Technical description:

(三張 test images: “baboon.jpg” 、 “peppers.jpg” 、

“pool.jpg” 需與 hw3.py 存在同個資料夾下，此作業使用了 python 的 OpenCV 套件)。

1. 定義執行 Sobel 邊緣檢測的函數，這個函數接受一個彩色影像作為輸入。首先，將輸入影像轉換為灰度影像，這是因為 Sobel 算子通常應用於灰度影像。接下來，使用 Sobel 算子在 x 和 y 方向上進行邊緣檢測。然後，計算梯度的大小，並將其歸一化到 0-255 的範圍內。最後 return 邊緣檢測後的影像。

```
1  import cv2
2
3  # 定義執行Sobel邊緣檢測的函數
4  def edge_detection(image):
5      # Convert the image to grayscale
6      gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7
8      sobel_x = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
9      sobel_y = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
10     # Calculate the magnitude of the gradient
11     magnitude = cv2.magnitude(sobel_x, sobel_y)
12     # Normalize the magnitude to a range of 0-255
13     magnitude = cv2.normalize(magnitude, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U)
14
15     return magnitude
16
```

2. 載入三張彩色影像，並對三張影像進行邊緣檢測，最後顯示出三個處理過後的 images

```
image1 = cv2.imread('baboon.png')
image2 = cv2.imread('peppers.png')
image3 = cv2.imread('pool.png')

edges1 = edge_detection(image1)
edges2 = edge_detection(image2)
edges3 = edge_detection(image3)

cv2.imshow('Image 1: baboon.png Edges', edges1)
cv2.imshow('Image 2: peppers.png Edges', edges2)
cv2.imshow('Image 3: pool.png Edges', edges3)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

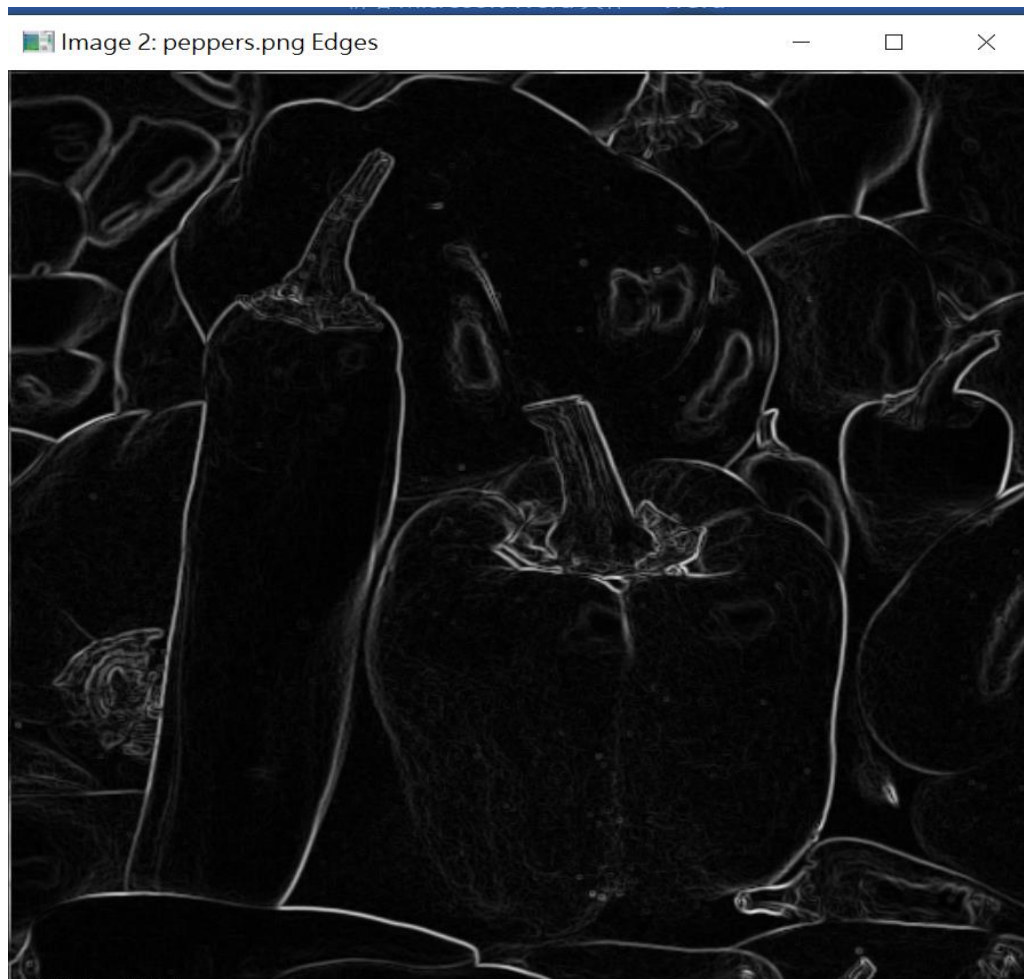
- Experimental results:

該程式的執行結果總共為三個輸出視窗

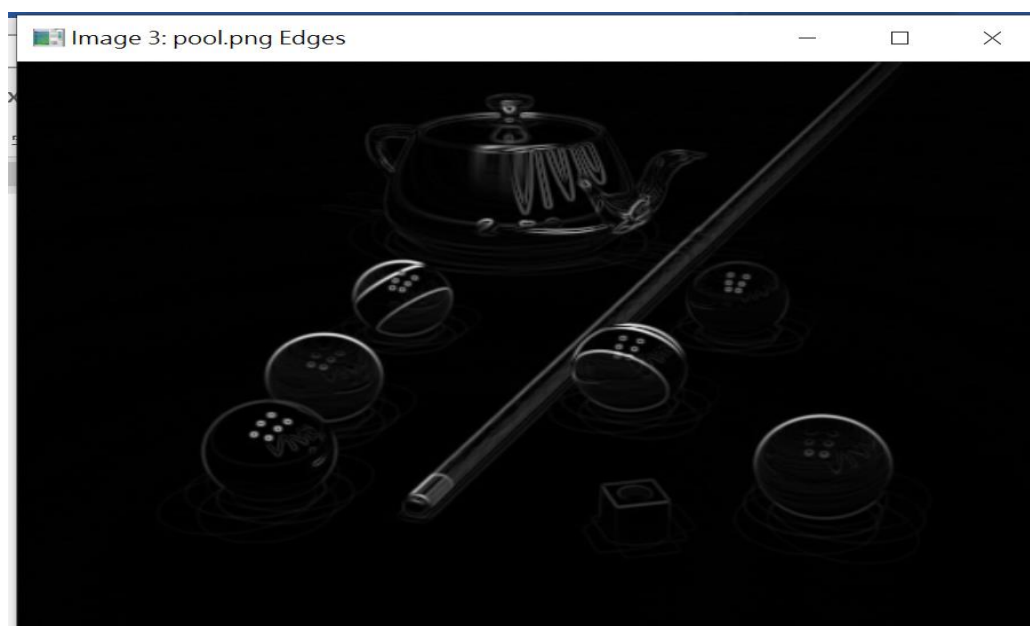
1. “Image 1: baboon.png Edges”視窗(如下圖)



2. “Image 2: peppers.png Edges”視窗(如下圖)



3. “Image 3: pool.png Edges”視窗(如下圖)



● Discussions:

Sobel算子結合了水平和垂直方向的梯度來檢測邊緣。Sobel算子的運作過程：將彩色影像轉換為灰度影像。這是因為Sobel算子通常應用於灰度影像，透過將彩色影像轉換為灰度，我們可以專注於亮度變化而不受顏色的影響。對灰度影像應用Sobel算子。這涉及兩個運算核，一個用於計算x方向上的梯度，另一個用於計算y方向上的梯度。這兩個核心是3x3的矩陣，對圖像中的每個像素進行運算，以計算該像素的梯度值。

根據x和y方向上的梯度計算梯度的大小。可以通過將x和y方向上的梯度結果平方後相加，然後取平方根得到梯度的大小。歸一化梯度大小。將梯度的大小歸一化到0-255的範圍，以便更好地顯示或進一步處理。

邊緣檢測的結果是一幅灰度影像，其中邊緣區域呈現高亮度，而非邊緣區域則較暗。邊緣檢測幫助我們識別圖像中的物體輪廓、形狀或結構，為後續的影像分割、物體識別等應用提供基礎。在程式碼中，我們對三張彩色影像分別應用了Sobel算子，並獲得了邊緣檢測的結果。這些結果存儲在edges1、edges2和edges3變數中，並使用cv2.imshow()函數顯示在屏幕上。

當我們顯示邊緣檢測影像時，邊緣區域將呈現明亮的白色或灰

色，而非邊緣區域將呈現較暗的顏色。這樣的視覺效果使我們可以清楚地看到影像中的邊緣部分，有助於進一步的影像分析和處理。

透過此次作業題目，我學習到Sobel算子的原理和應用，以及如何使用它來進行邊緣檢測，了解Sobel算子在x和y方向上的運算核心，以及如何計算梯度大小。也學習到邊緣檢測的概念和原理，了解邊緣檢測在計算機視覺和影像處理中的重要性，以及它在物體檢測、輪廓識別等應用中的作用。

● References and Appendix:

--references:

1. https://www.tutorialspoint.com/dip/sobel_operator.htm
2. <https://github.com/tharunchitipolu/sobel-edge-detector>
3. <https://onlinedocs.microchip.com/g/GUID-37AD5EEE-6FAB-48FC-89F6-CAA649534B2A-en-US-1/index.html>