

Homework#1

姓名：黃楚祐

學號：00557043

日期：2019/3/30

方法

1. `adaptiveThreshold`: 試著把魚和水的灰度二值化，分辨出畫面中的魚和水。

```
gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY) #frame為cap.read()的到的影像
#要先轉成灰階影像才能二值化
biframe = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 163, 8)
#最後兩個參數為 blockSize=163, threshold = mean - 8
frameFull[0:frameHeight, frameWidth:, 0] = biframe
frameFull[0:frameHeight, frameWidth:, 1] = biframe
frameFull[0:frameHeight, frameWidth:, 2] = biframe ##biframe剩下一個channel 所以輸出BGR的時候三個都要填
```

2. `backgroundSubtractorMOG2`: 藉由當前的 `frame` 與前一個 `frame` 相減，讓靜止的背景與運動中的前景二值化，分離出靜止的水、波動的水、魚。

```
gbg = fgbg.apply(frame) #fgbg = cv2.createBackgroundSubtractorMOG2() ##cv2的此方法不能調參數
frameFull[frameHeight:, 0:frameWidth, 0] = gbg
frameFull[frameHeight:, 0:frameWidth, 1] = gbg
frameFull[frameHeight:, 0:frameWidth, 2] = gbg ##最後也會二值化，所以剩下一個channel，BGR都要填值
```

3. 畫出由左上到右下先變大再縮小的長方形，RGB 在每個 `frame` 都會 `random` 一次。

```
def draw_rectangular(img, n, total, height, width): #當前的影像, 當前幀數, 總幀數, 影像高度, 影像寬度
    color1 = int(random.random() * 256)
    color2 = int(random.random() * 256)
    color3 = int(random.random() * 256) #BGR顏色random
    ratio = n/total #算出目前的幀數與總幀數的比例
    if ratio < 0.5: #影片到一半的時候矩形全滿
        h = int(ratio*height)*2;
        w = int(ratio*width)*2;
        img[0:h,0:w] = [color1, color2, color3]
    else: #影片超過一半的時候，矩形從左上開始變小
        ratio = ratio - 0.5
        h = int(ratio*height)*2;
        w = int(ratio*width)*2;
        img[h:height, w:width] = [color1, color2, color3]
    return img
```

結果

我用 Adaptive threshold 處理這支影片的方法不太理想，因為水的波紋變化太多，導致產生很多干擾，並且花了好多時間調參數。

BackgroundSubtractorMog2 還是會被變化太快的水波紋干擾，但如能調整程式，讓變化過大的地方也變成黑色，那麼魚的部分會更清楚。

影片: <https://www.youtube.com/watch?v=uqSozDmMW94&feature=youtu.be>

結論

這次我寫的作業大部分都是呼叫 cv2 裡的 function 調參數，但我也學到跑兩層迴圈對二維陣列填入一樣的值不如使用 frame[0:n, 0:n]來的快。相信往後的作業能讓我學到更多使用 python 與 openCV 的技巧。

我的眼睛和大腦，比我目前學到或想到的影像處理演算法厲害多了，眼見 2、30 隻魚在影片裡游泳，但經過影像處理後能清楚辨識的不過 10 隻。該如何增加辨識的精準度?與該如何接續處理已處理的影像?

這些問題驟增我對機器視覺的興趣，經過第一次作業得知機器視覺的確是大工程。

參考文獻

OpenCV 對 Background Subtraction 的介紹

https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html#py-background-subtraction