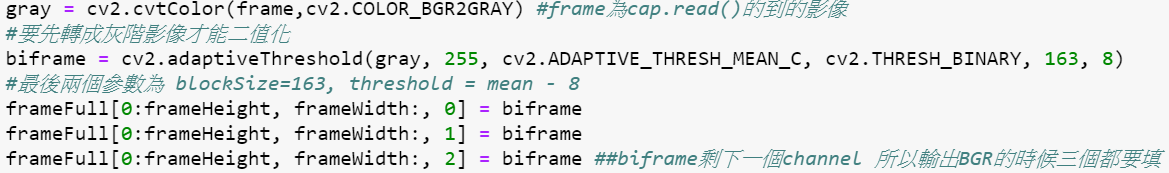
|  |
| --- |
| 機器視覺作業報告 |
| Homework#1 |
|  |
|  |
| 姓名: 黃楚祐  學號: 00557043  日期: 2019/3/30 |
|  |

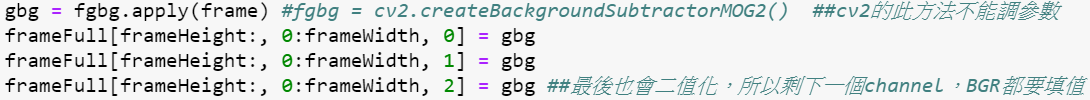
|  |
| --- |
|  |

# 方法

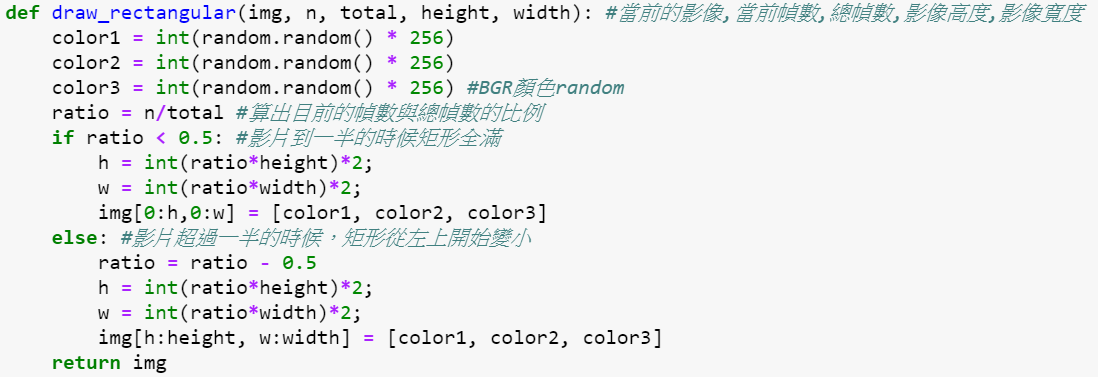
1. adaptiveThreshold: 試著把魚和水的灰度二值化，分辨出畫面中的魚和水。



1. 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。



1. 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。



# 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>