RPi OPENCV範例實作 (Ref:CAVEDU袁)

1.請打開RPi內建端終機，依序鍵入下列命令，以安裝需要的套件

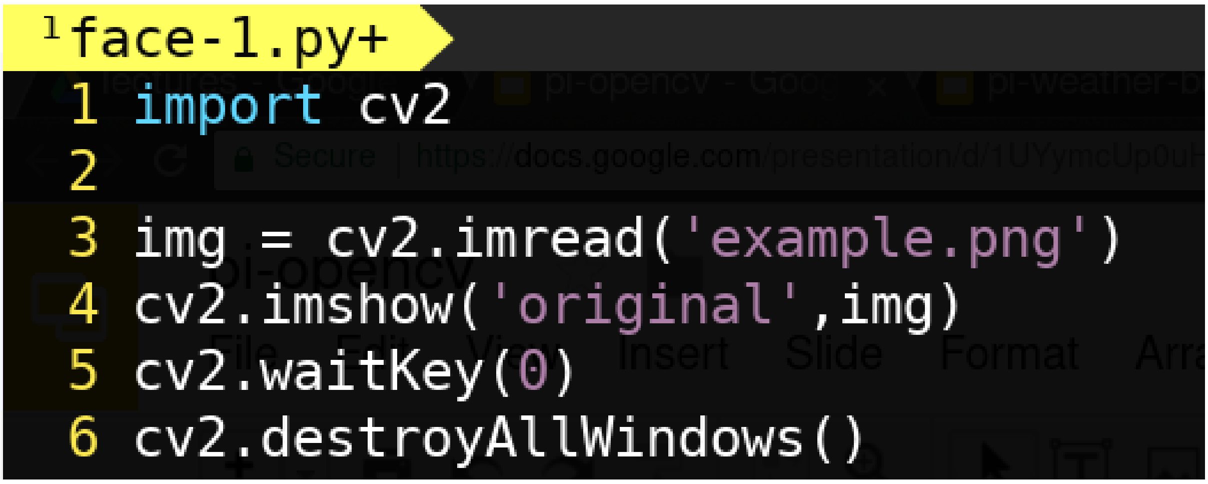
```
sudo apt-get update
```

```
sudo apt-get install
    libopencv-dev
```

```
sudo apt-get install python3-opencv
```

2.開始進行範例1編輯執行，主要動作為讀入照片顯示

$nano face-1.py



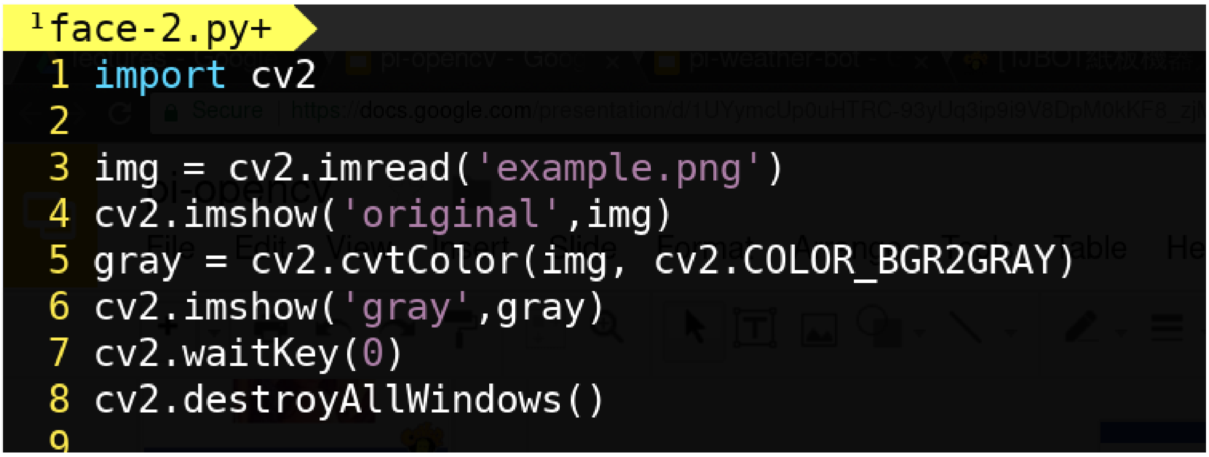
```
import cv2

img = cv2.imread('example.png')
cv2.imshow('original',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

存檔Ctrl+o, 按Yes , Ctrl+x退出

$python face-1.py 〈--按Enter後執行

其他範例程式，以此類推進行編輯執行動作

範例2 主要為接續範例1再進行照片的灰階化動作



```python
import cv2

img = cv2.imread('example.png')
cv2.imshow('original',img)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('gray',gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

CV2常見影像格式轉換函式功能

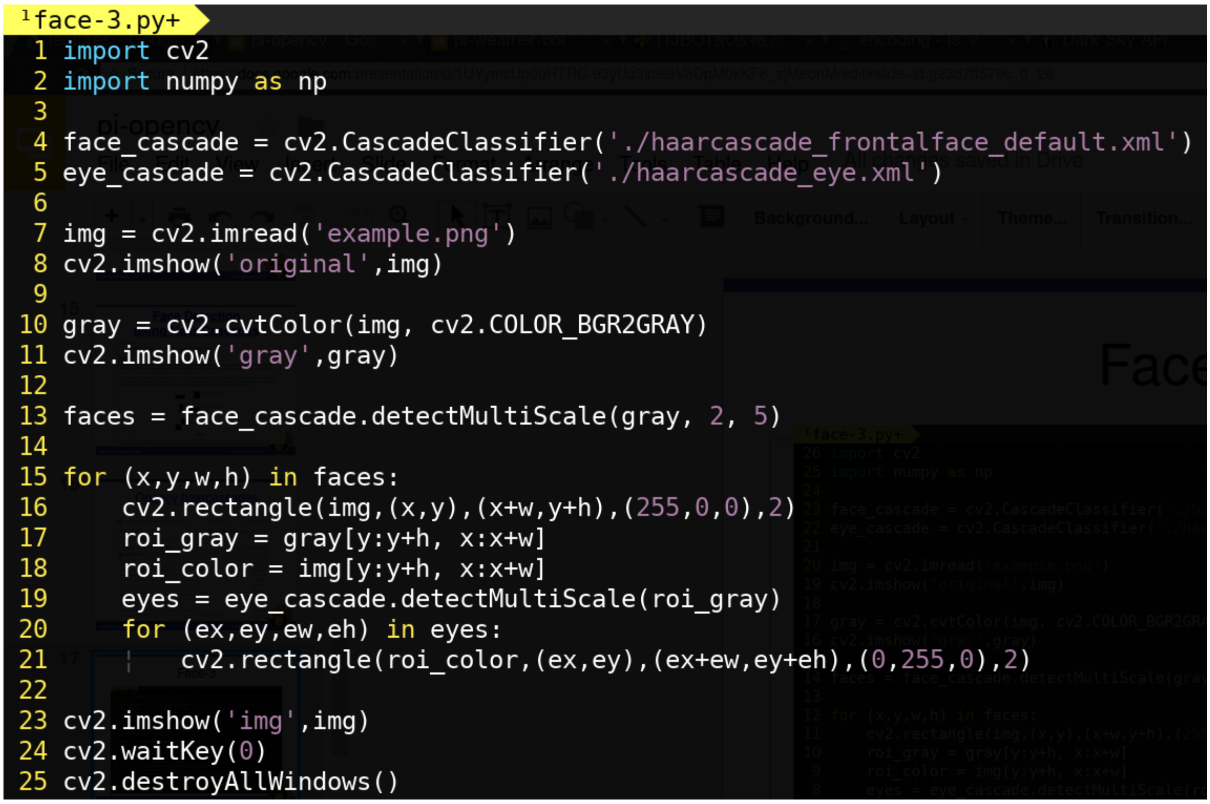
cv2.COLOR_BGR2GRAY #RGB轉**灰階**

cv2.COLOR_BGR2HSV #RGB轉HSV

cv2.COLOR_HSV2BGR #HSV轉RGB

範例3 :進行人臉識別及標定

$ nano face-3.py



```python
import cv2
import numpy as np

face_cascade = cv2.CascadeClassifier('./haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('./haarcascade_eye.xml')

img = cv2.imread('example.png')
cv2.imshow('original',img)

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('gray',gray)

faces = face_cascade.detectMultiScale(gray, 2, 5)

for (x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
$ wget https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml
$wget https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_eye.xml
$ python face-3.py
```

haarcascade特徵分類器用途, xml檔存放既有特徵值比對參考

frontalface_default.xml->人臉（正，側面）

frontalface_alt2.xml->以正面人臉為主

frontalface.xml ->以側面人臉為主

frontalface_eye.xml->以眼睛為為主

注意參數使用

畫出矩陣方塊 CV2.rectangle(格式, 左上座標，右下座標，顏色，線寬)

ROI->特定區域處理

detectMultiScal(image,scaleFactor,minNeighbors)

haar feature cascade (scaleFactor , minNeighbors , minSize)

scaleFactor:檢測視窗縮放比率，值愈小,較能偵測人臉。

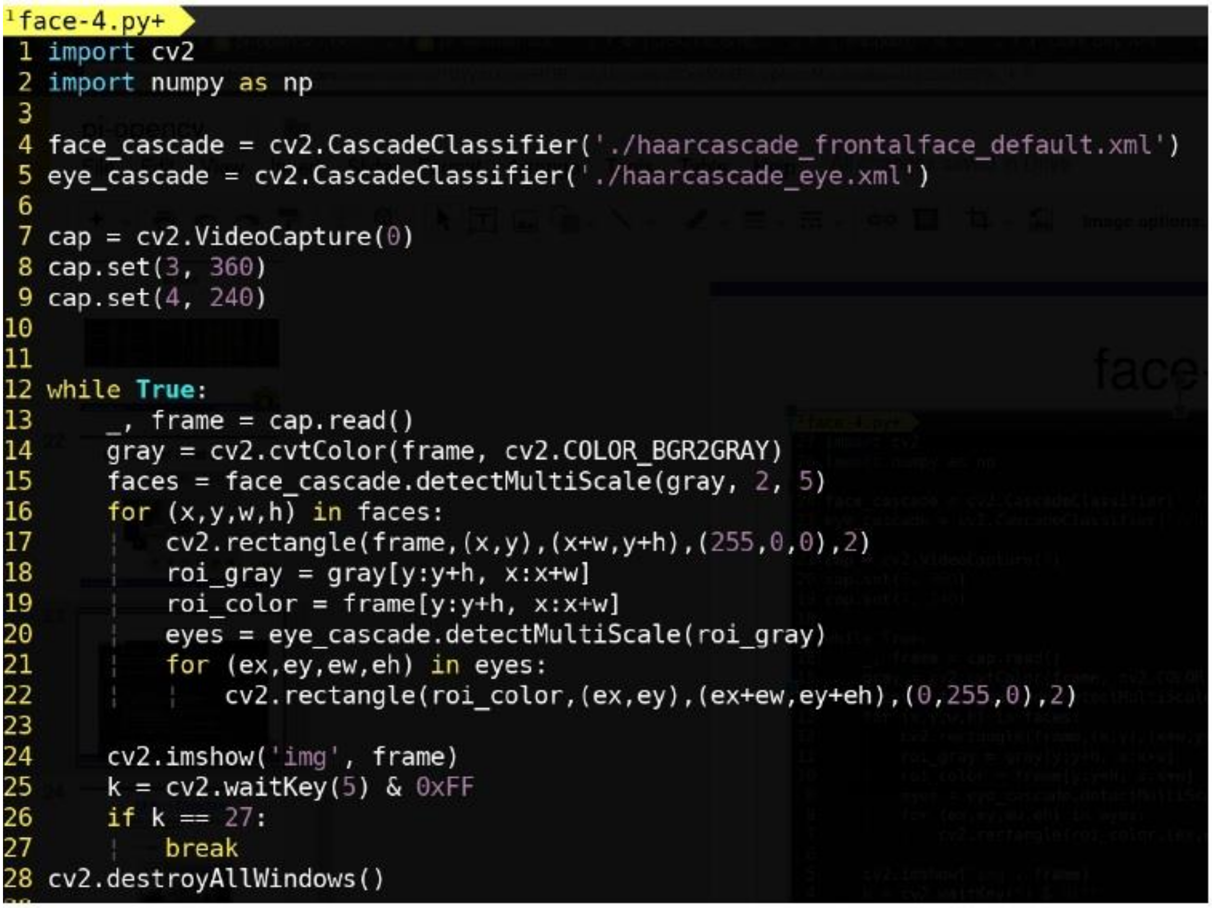
minNeighbors:檢測鄰近內最小檢測出的5人臉次數

minSize:被檢測物體的最小尺寸

影像辨識流程：

取樣->訓練-〉識別

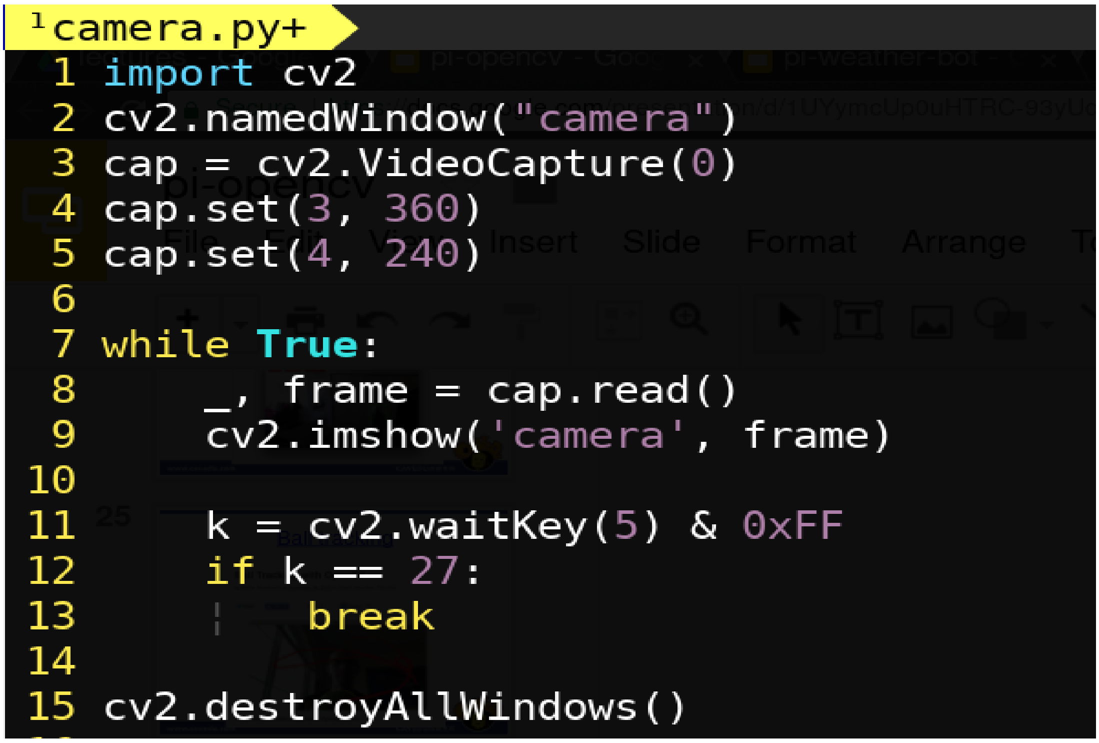
- 插入WebCAM C170 USB接頭至RPi，確認RPi是否可以識別到
- $lsusb



face-4.py+

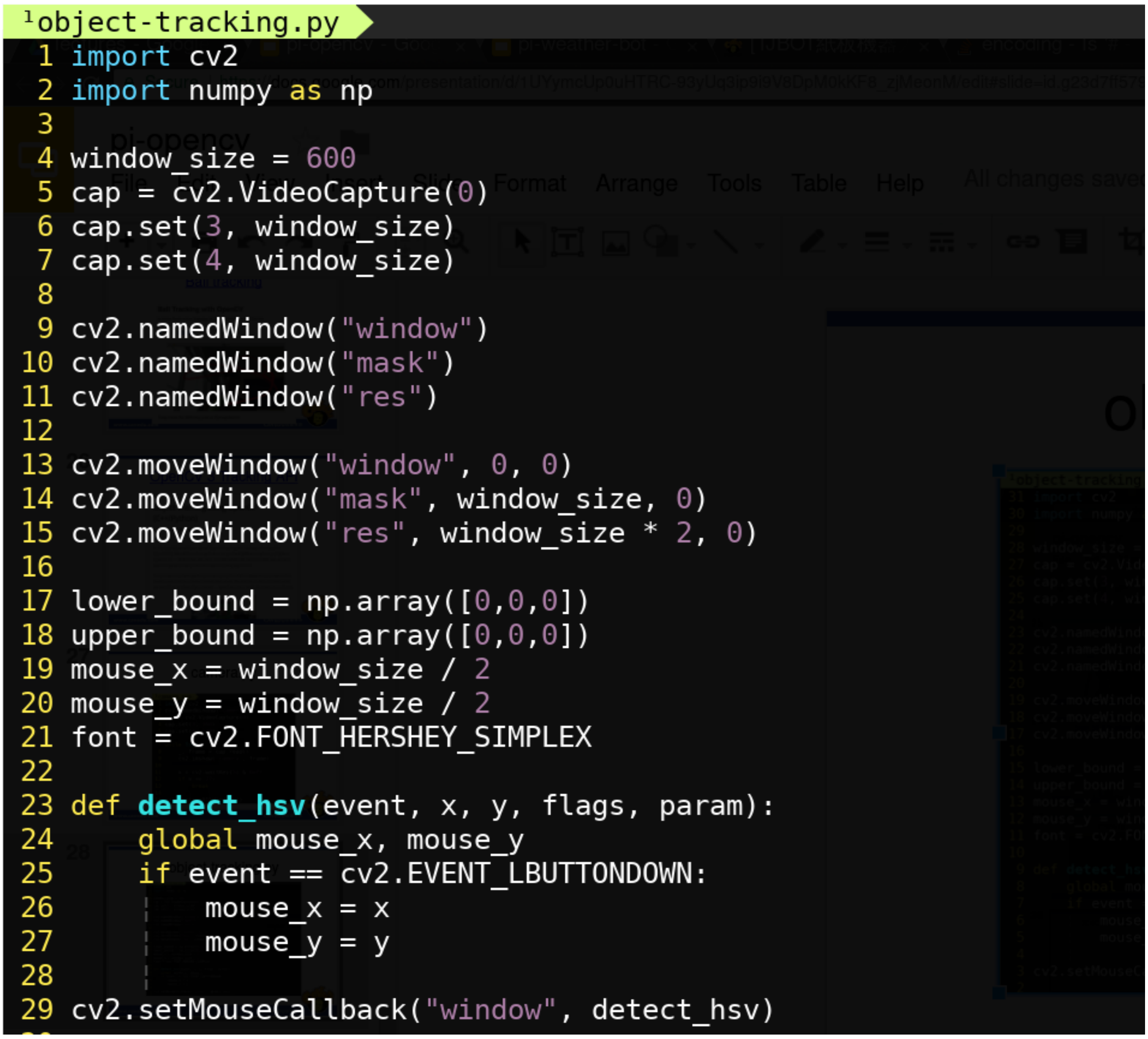
```python
import cv2
import numpy as np

face_cascade = cv2.CascadeClassifier('./haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('./haarcascade_eye.xml')

cap = cv2.VideoCapture(0)
cap.set(3, 360)
cap.set(4, 240)


while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 2, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

    cv2.imshow('img', frame)
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break
cv2.destroyAllWindows()
```

範例4：串流2影像

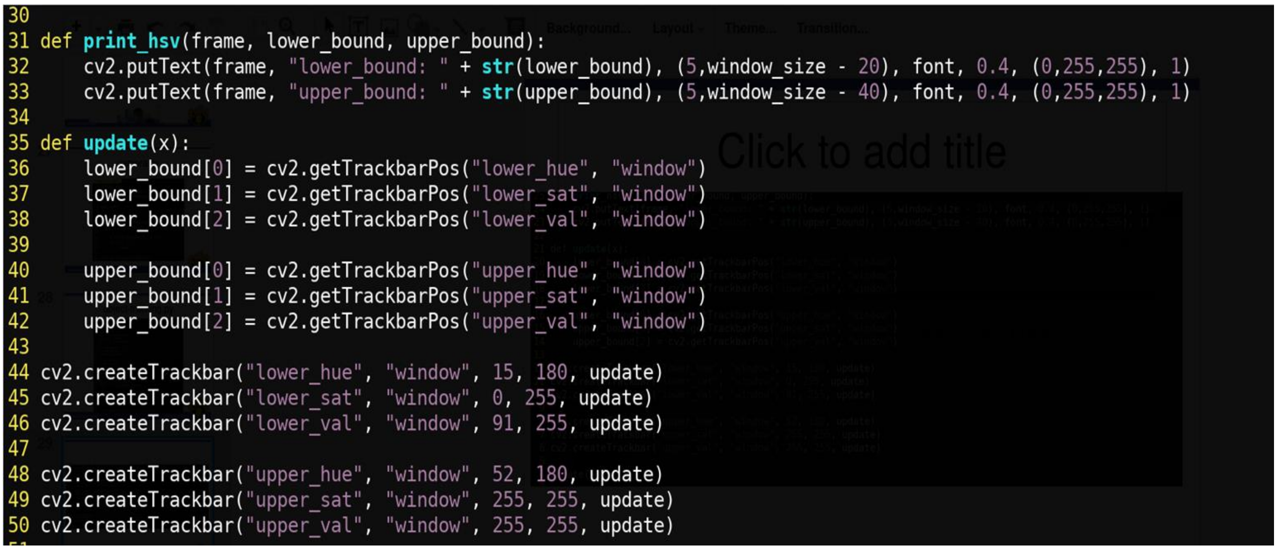


```python
import cv2
cv2.namedWindow("camera")
cap = cv2.VideoCapture(0)
cap.set(3, 360)
cap.set(4, 240)

while True:
  _, frame = cap.read()
  cv2.imshow('camera', frame)

  k = cv2.waitKey(5) & 0xFF
  if k == 27:
    break

cv2.destroyAllWindows()
```

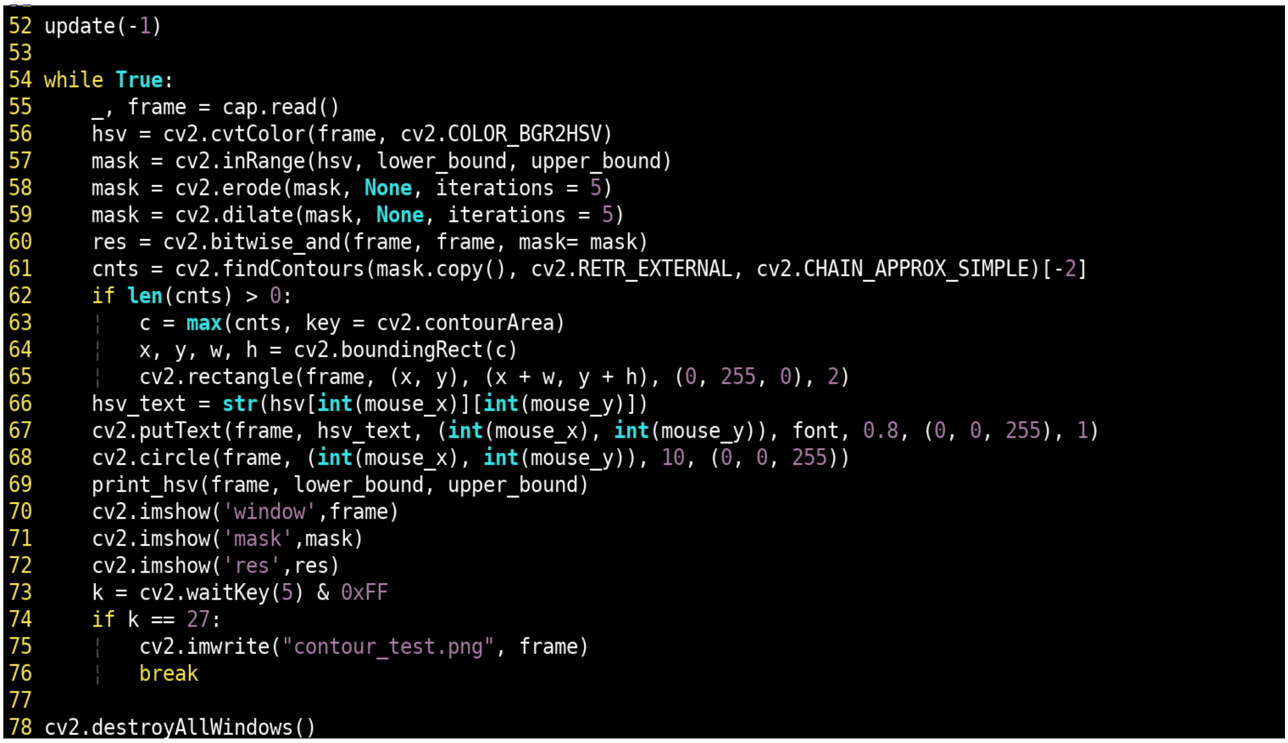
範例5：物件追蹤



```python
import cv2
import numpy as np

window_size = 600
cap = cv2.VideoCapture(0)
cap.set(3, window_size)
cap.set(4, window_size)

cv2.namedWindow("window")
cv2.namedWindow("mask")
cv2.namedWindow("res")

cv2.moveWindow("window", 0, 0)
cv2.moveWindow("mask", window_size, 0)
cv2.moveWindow("res", window_size * 2, 0)

lower_bound = np.array([0,0,0])
upper_bound = np.array([0,0,0])
mouse_x = window_size / 2
mouse_y = window_size / 2
font = cv2.FONT_HERSHEY_SIMPLEX

def detect_hsv(event, x, y, flags, param):
    global mouse_x, mouse_y
    if event == cv2.EVENT_LBUTTONDOWN:
        mouse_x = x
        mouse_y = y

cv2.setMouseCallback("window", detect_hsv)
```



```python

def print_hsv(frame, lower_bound, upper_bound):
    cv2.putText(frame, "lower_bound: " + str(lower_bound), (5,window_size - 20), font, 0.4, (0,255,255), 1)
    cv2.putText(frame, "upper_bound: " + str(upper_bound), (5,window_size - 40), font, 0.4, (0,255,255), 1)

def update(x):
    lower_bound[0] = cv2.getTrackbarPos("lower_hue", "window")
    lower_bound[1] = cv2.getTrackbarPos("lower_sat", "window")
    lower_bound[2] = cv2.getTrackbarPos("lower_val", "window")

    upper_bound[0] = cv2.getTrackbarPos("upper_hue", "window")
    upper_bound[1] = cv2.getTrackbarPos("upper_sat", "window")
    upper_bound[2] = cv2.getTrackbarPos("upper_val", "window")

cv2.createTrackbar("lower_hue", "window", 15, 180, update)
cv2.createTrackbar("lower_sat", "window", 0, 255, update)
cv2.createTrackbar("lower_val", "window", 91, 255, update)

cv2.createTrackbar("upper_hue", "window", 52, 180, update)
cv2.createTrackbar("upper_sat", "window", 255, 255, update)
cv2.createTrackbar("upper_val", "window", 255, 255, update)
```



```python
update(-1)

while True:
    _, frame = cap.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, lower_bound, upper_bound)
    mask = cv2.erode(mask, None, iterations = 5)
    mask = cv2.dilate(mask, None, iterations = 5)
    res = cv2.bitwise_and(frame, frame, mask= mask)
    cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[-2]
    if len(cnts) > 0:
        c = max(cnts, key = cv2.contourArea)
        x, y, w, h = cv2.boundingRect(c)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    hsv_text = str(hsv[int(mouse_x)][int(mouse_y)])
    cv2.putText(frame, hsv_text, (int(mouse_x), int(mouse_y)), font, 0.8, (0, 0, 255), 1)
    cv2.circle(frame, (int(mouse_x), int(mouse_y)), 10, (0, 0, 255))
    print_hsv(frame, lower_bound, upper_bound)
    cv2.imshow('window',frame)
    cv2.imshow('mask',mask)
    cv2.imshow('res',res)
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        cv2.imwrite("contour_test.png", frame)
        break

cv2.destroyAllWindows()
```

open_cv動態範例

http://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/

http://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/