

Lab08

目標

- 利用HOG行人檢測及Dlib臉部偵測框出人(25%)與人臉 (25%)
- 利用任一方法算出與其的距離
- demo時為即時影像並用尺量人(25%)與人臉 (25%) 距離準確度
- demo誤差: 人(50cm)、人臉(10cm)

HOG(Histogram of Oriented Gradient)

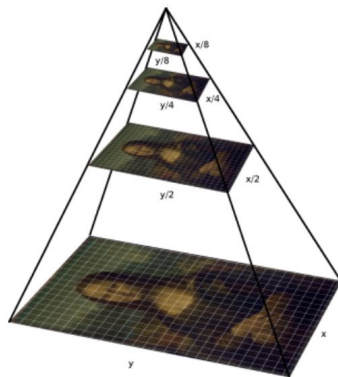
initialize the HOG descriptor/person detector

- `hog = cv2.HOGDescriptor()`
- `hog.setSVMDetector(cv.HOGDescriptor_getDefaultPeopleDetector())`
- `rects, weights = hog.detectMultiScale(src, #輸入圖`

`winStride`, #在圖上抓取特徵時窗口的移動大小

`scale`, #抓取不同scale (越小就要做越多次)

`useMeanshiftGrouping = False)`



Dlib Face Detection

(python >=3.7)

pip install cmake

pip install dlib

Dlib Face Detection

- `import dlib`
- `detector = dlib.get_frontal_face_detector()`
- `face_rects = detector(img, 0)`
- 取出所有偵測的結果
 - for `i, d` in `enumerate(face_rects)`:
 - `x1 = d.left()`
 - `y1 = d.top()`
 - `x2 = d.right()`
 - `y2 = d.bottom()`

畫出長方形

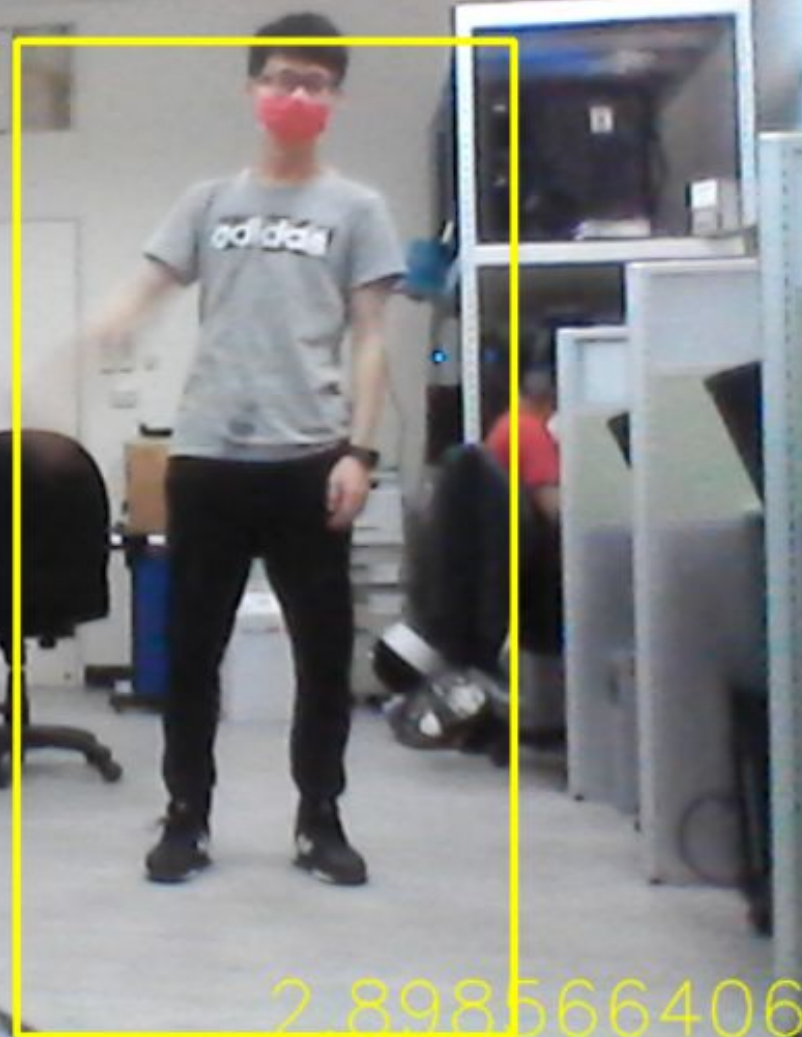
- `image = cv2.rectangle(image, start_point, end_point, color, thickness)`

深度預測

- 不限定方法

1. 已知物體大小及相機焦距，用物體在畫面中占的pixel計算
物件的框會有留白，可以自行判斷要乘多少比例才是物體實際pixel大小
2. 假設人或人臉為平面，已知大小解SolvePnP
 - `cv2.solvePnP(objp, imgPoints, intrinsic, distortion)→ retval, rvec, tvec`
objp的部分要用真實的長度單位，非(0,0), (0,1), (1,0), (1,1)

[54 94645948]



2.898566406