Lab08

目標

- 利用HOG行人檢測及Haar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

用real time

抓64*128 -> 要放大的話把圖片縮小

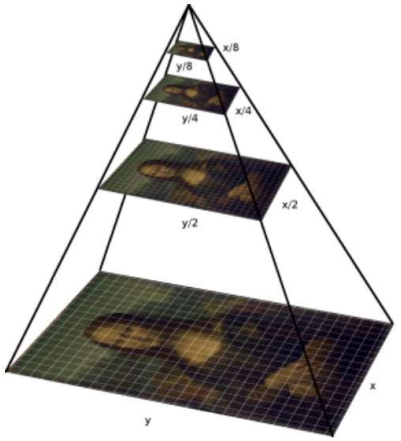
HOG(Histogram of Oriented Gradient)

 # initialize the HOG descriptor/person detector

- hog = cv2.HOGDescriptor()　建HOG特徵描述
- 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)

影響速度和精準度

每次縮放大小
scale越小越精準
但速度越慢



Haar-cascade Face Detection

- face_cascade = cv2.CascadeClassifier('opencv/data/haarcascades/haarcascade_frontalface_default.xml')

  存在opencv的資料夾

- faces = face_cascade.detectMultiScale(
  src, , #輸入圖
  scaleFactor , #抓取不同scale
  minNeighbors, #該區域附近被認為是臉的次數
  minSize, #物體最小限制
  maxSize, #物體最大限制
)

畫出長方形

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

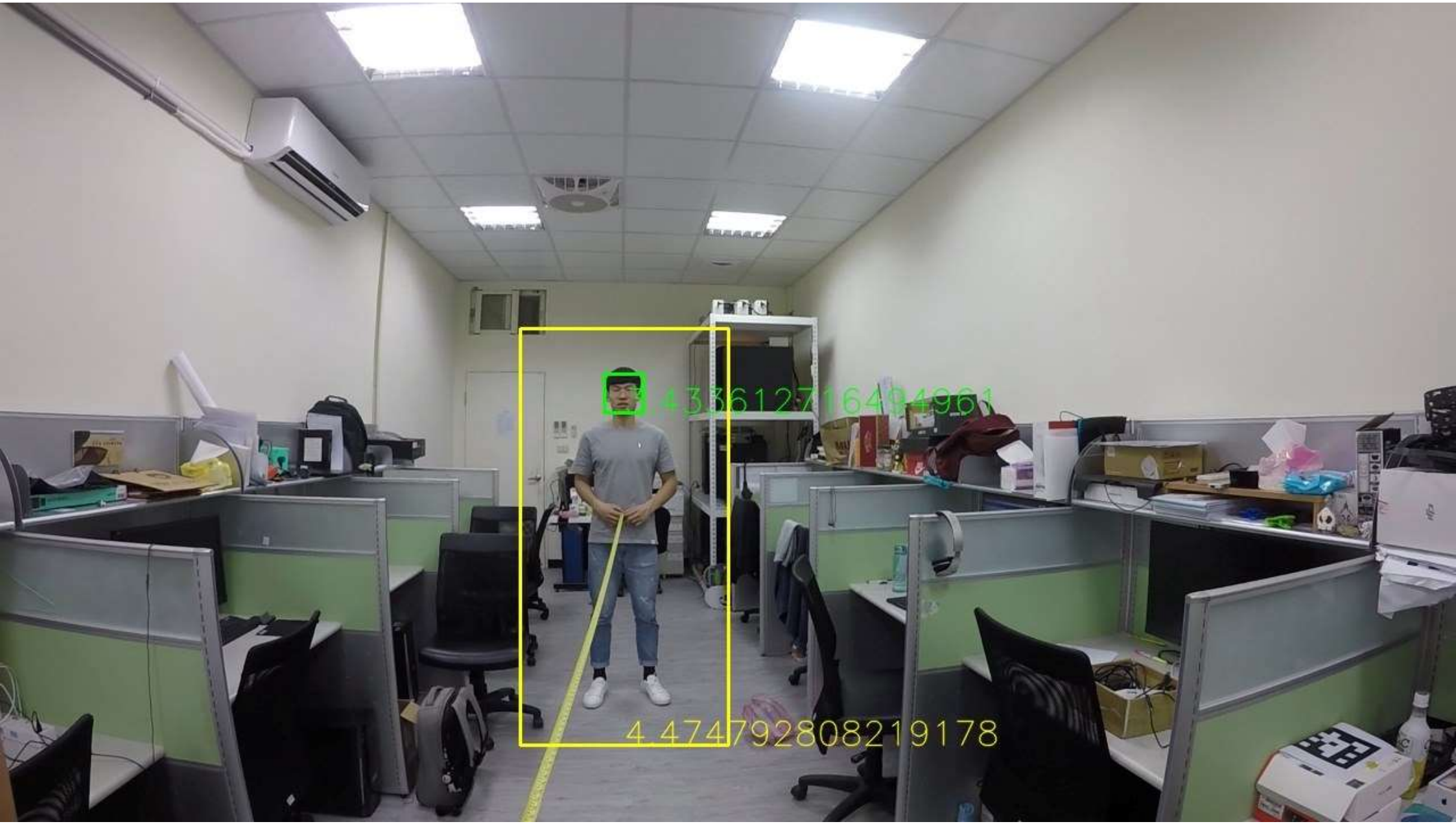
深度預測

- 不限定方法

1. 腳跟地面交點來算距離　打人的腳看其射線
2. 已知高度　相似三角形
3. 假設人或人臉為平面, 已知大小解SolvePnP 假設物件是平面，不建議這樣做

- cv2.solvePnP(objectPoints, imagePoints, cameraMatrix, distCoeffs[, rvec[, tvec[, useExtrinsicGuess[, flags]]]]) → retval, rvec, tvec

上下都有間隔，估的時候記得扣一點



偵測call function就好，難的是估距離