

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(100cm)、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`

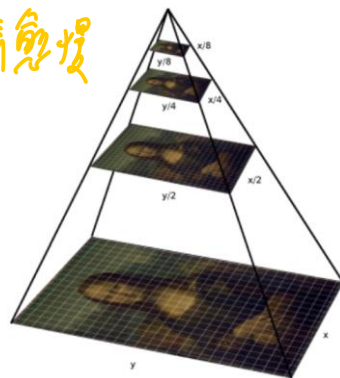
`winStride`, #在圖上抓取特徵時窗口的移動大小

`scale`, #抓取不同scale (越小就要做越多次) → 會越準, 但會愈慢

`useMeanshiftGrouping = False`)

偵測到行人
的話
會回傳行人
的4個值

偵測大於可能度的人



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()
- Handwritten notes in blue:
- ↑ 偵測到的面 (points to `face_rects`)
 - ↓ 拿到 detector (points to `detector`)
 - ↑ 圖片有幾張 (points to `0` in `detector(img, 0)`)

畫出長方形

左上
↓

右下的點都連到
↓

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

深度預測

知道 webcam 高度，

用相似 Δ 算人臉和鏡頭的距離

- 不限定方法

1. 已知高度

2. 假設人或人臉為平面，已知大小解SolvePnP → open cv 的 function

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

3D 點

同一個物件的 2D 點



433612716494961



4.474792808219178