

Lab08

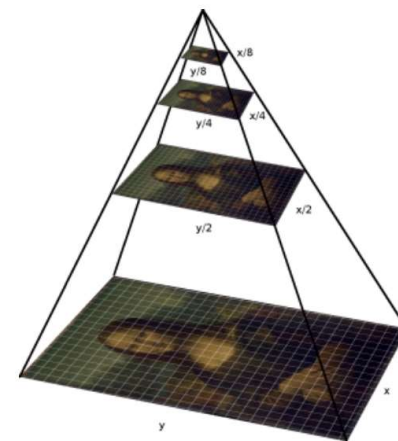
目標

- 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

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)`



Haar-cascade Face Detection

- `face_cascade =`
`cv2.CascadeClassifier('opencv/data/haarcascades/haarcascade_frontalface_default.xml')`
- `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
 2. 已知高度
 3. 假設人或人臉為平面, 已知大小解SolvePnP
- `cv2.solvePnP(objectPoints, imagePoints, cameraMatrix, distCoeffs[, rvec[, tvec[, useExtrinsicGuess[, flags]]]])` → `retval, rvec, tvec`

