

一、Real-time Facial Recognition

Facial Recognition 分成以下人臉偵測與人臉辨識兩部分說明

1. 人臉偵測：

偵測的部份我們使用 opencv3.4.7 裡面就有的 Haar 進行偵測，很方便，只要 load "haarcascade_frontalface_default.xml" 就可以使用了。

2. 人臉辨識：

我們使用 LBPH 去進行辨識，這個 module 不包含在 OpenCV 主套件中，要到 [opencv github](https://github.com/opencv/opencv_contrib) 載 **opencv_contrib** module source code，載好後放到 cmake-gui 中的 OPENCV_EXTRA_MODULES_PATH，並且至少需要關掉 BUILD_opencv_cvv module，不然 compile 的時候會報錯，接著就是 compile 出 opencv 的 .so 檔案，就可以使用 LBPH 了。

LBPH 辨識前需要先進行訓練，所以我們用 Lab2 的截圖功能截了很多張我們的臉，接著用這些照片去訓練 LBPH。

訓練的時候會先使用 Haar 將人臉區域裁切出來，再交由 LBPH 計算該張臉的 LBP 特徵向量，每個人會對應到一組特徵向量並存入資料庫。

人臉辨識流程：

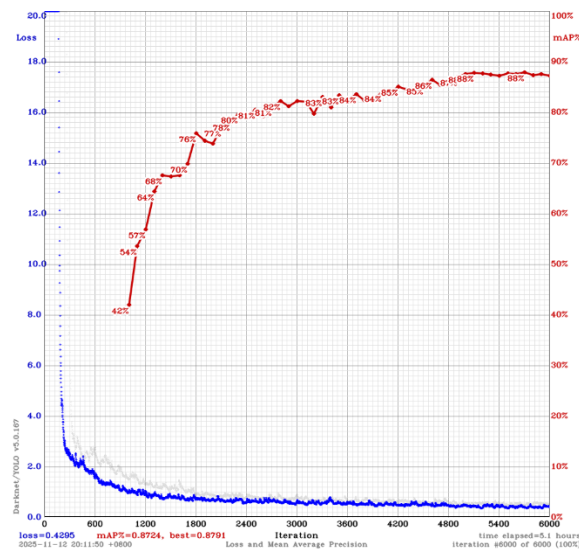
1. 將 input 轉成灰階。
2. 丟給 Haar 進行人臉偵測，取得人臉的 bounding box。
3. 將 bounding box 中的人臉裁切下來並轉成灰階。
4. 丟給 LBPH 計算出 LBPH 特徵向量。
5. 最後，與資料庫中的 LBPH 特徵向量做距離比對，若計算出來的 confidence 低於 80 則去標上那個人的學號，若高於 80，表示很沒自信，就標為 Unknown。
(confidence 是距離，**越低越好**，越低表示越相似)

二、Helmets Detection

安全帽偵測的話我們使用 YOLO 去進行偵測，本來是使用 YOLOv5 並希望能用 opencv 裡面的 dnn module 去接模型，但是嘗試後發現 v5 太新了 opencv 太舊了，所以會有問題。因此我們又嘗試將訓練好的 YOLOv5 轉成 ncnn 與 pnnx，再讓 dnn 去接，但是最後還是有 floating point 的問題。最後，我們就去訓練了 YOLOv3 的模型，因為 opencv 的 dnn 確定可以讀取 YOLOv3 的 darknet。

訓練的部分，主要都是從 kaggle 與 roboflow 中下載 Helmet Dataset 去訓練我們的模型，設定 class 只有一個，就是 Helmet。但是 YOLOv3 需要 label 是 txt 檔案，而 kaggle 下載下來的 dataset 都是 xml，所以我們用 python 去將 xml 轉成 txt。

以下是我們 YOLOv3 訓練的 loss 曲線圖：



接著要部署到板子上，首先用”dnn::readNetFromDarknet”將 YOLOv3 的結構跟訓練好的權重 load 進去，接著用 forward() 執行推論，他會將偵測出來的所有 bounding box 與其的分數都儲存起來，最後就是看每個分數有沒有高於我們設定的 confidence threshold，若高於的話就在原圖上畫框並標上 "Helmet"。