

專題名稱: Selfie Helper 自拍偵測結合透過偵測手部動作自動套用/

切換/取消不同種類濾鏡

1. 動機

- 取代自拍時需手動設定倒數計時的困擾
- 出門前自拍棒總會忘記充電，在旅程中自拍棒沒電會造成困擾，透過自拍偵測即不須任何操作即可倒數拍照
- 拍照前選擇濾鏡總需動手動設定選擇很久，選完濾鏡後實際套用到臉上可能也不一定合適，又要重新選。此系統實現預先設定幾種自己常用的濾鏡，鏡頭直接面對臉部下可直接透過手部姿勢來更換預設的濾鏡，省去不少困擾
- 可事先設定多種濾鏡，拍照時直接套用，減少了之後一張一張修圖的困擾
- 使用 AI makeup 技術，免除修圖新手不擅長修圖之困擾
- 可透過特定手勢自動開始/終止自拍

2. 實作步驟

1. 尋找 selfie dataset 並訓練出 selfie detection model
2. 尋找 hand sign dataset 並訓練出 hand sign detection model
3. 使用 opencv 抓取鏡頭內照片
4. 使用 cvzone 之 PostDetector 抓取身體物件
5. 透過身體物件之數據計算出身體物件佔畫面之比例(在此稱 body_proportion)
6. 偵測到人物以及 body_proportion 符合則進行 selfie detect，符合則倒數自拍
7. 使用 cvzone 之 HandDetector 抓取手部物件
8. 在倒數階段，若手筆出特定數字之手勢，則可切換至該 mode 之濾鏡
9. 修改現成之 AIMakeUp 程式，即時套入臉部濾鏡
10. 整合

3. 訓練之模組: Selfie Detection model 和 Hand Sign Detection model 皆使用

CNN 訓練，詳細過程以及說明皆在 Selfie_Helper.ipynb 裡面

4. 重要套件:

- 使用 PyQt5 實作 GUI
- 使用 dlib 對人臉實作 AI Makeup
- 使用 cvzone 的 HandDetector 實現手部抓取
- 使用 cvzone 的 PoseDetector 實現手部抓取

- 使用 cvzone 的 SelfiSegmentation 實現去背景

5. 重要步驟說明

- AIMakeup 部分修改現成程式碼完成，並連接上 PyQt5 之 GUI
- 使用 Json 存下 5 種濾鏡，未來能套用在自拍的臉上
- 開啟鏡頭後，左上顯示的是倒數資訊，偵測到人物後，計算並顯示出人物佔畫面比例，到達一定比例後會，會抓取人物照片特定部分進行 **resize**，**reshape**，再用照片去預測是否在 **selfie**，並在人物框上面顯示狀態。預測為正在自拍時，便會開始倒數計時
- 開啟鏡頭後便開始抓取手部位置，擷取出手部照片位置，轉換為灰階，去背景，**resize**，**reshape**，再加上再將手部照片外框加厚調整手部比例，才拿去預測手部動作，動作為 0 為停止自拍倒數，9 為開始自拍倒數，1~5 為套用預先設定好的 5 種濾鏡

6. 心歷路程

一開始是決定使用年齡預測模組來延伸專題，但訓練完後預測非常不准，改善模組也沒甚麼效果，之後想了想得知可能一方面是資料集問題，**沒辦法便是各種國家的面孔，一方面是連人類都沒辦法準確辨認年齡，機器學習訓練出來的當然會更不準確**

接著便找到了一個 selfie 的資料集，看 kaggle 上沒有人有使用過這資料集來訓練 model，也沒有其他 selfie 相關的資料集，好像也不是熱門的機器學習主題，但我想嘗試一下，覺得可以從這 model 來延伸出 selfie helper 的程式。接著又找到了手部姿勢的資料集，但其實一直找不到好的資料集，找到了之後訓練完發現一個問題：**每個國家的手勢代表的意義其實不一樣**，最後我便只取資料集中的比較通用的 0~5 還有辨識度比較高的 9 來做訓練。訓練出 **selfie detection model** 後，準確率很高，模型評分也不錯，**但之後自己測照片卻沒有很理想，之後發現是資料集會有缺陷，人物比例較小的會被偵測沒在自拍**，要經過人物抓取後再抓取特定部位，才能較準確的預測

訓練 **hand sign detection model**，準確率很高，模型評分也不錯，但是實際套用後根本不准。**後來才發現也是手部比例問題，還有背景問題，以及訓練資料皆是灰階照片，故在訓練前我有先處理過大小，轉灰階，去背景，以及加厚外框調整手部**，才得到較準的預測。

但接著又發現一個問題：不同手勢的辨識準確度不一樣，像是'4'很容易辨

認，'1','9'很不容易辨認，故我採用以 10 次預測為單位的方法，10 次內有幾次是該數字，才預測為該數字，像是'4'需要 10 次內出現 5 次才會認為是'4'，而'1'需要出現 3 次便會認為是'1'。使用這方法過後準確度好了很多，同時也使敏感度降低，模式不會變換過快。

在使用 camera 時也碰到一個問題: 本來在 colab 使用 camera，但發現他抓取頻率過慢，導致會卡頓，而 opencv 的 camera 好像無法再 colab 上跑，故才轉為在 local 執行

尋找手部以及人物抓取套件時也花了不少時間，先前也使用了 opencv 的 harr cascade，但比較過後還是使用了 cvzone

接著也碰到一個問題: 同時加入 model 預測以及人物抓取，手部抓取，套用 AI Makeup 後，畫面延遲會變高，故我將在進入倒數階段時將人物抓取移除，自拍預測移除，畫面順了不少，但還是會因為 AIMakeup 時間過長造成 delay

7. 心得

一開始想主題找資料集花了非常久的時間，我目標使用 image 的資料集訓練，以及能夠延伸另一個 model 的，翻遍了 kaggle 才找到一個比較有靈感的資料集。

原本以為這專題花幾天就能解決了，沒想到整整花了兩個禮拜在這上面，本來規劃在畢業專題以及期末考讀書，研究所全部時間都花在這期末專題上了。雖然也曾想說只是一個幾周時間期末專題不用花那麼多心力，也不需要訓練到兩個資料及結合那麼多套件還有現有程式，但卻是越做越心血來潮，想要把它弄到最好。

在訓練的時候就花了很多時間重新訓練了，在選擇套件，camera 無法抓取 frame 的問題，model 實際精準度不理想的問題也都反惱了我很久，花了很多時間。但最後做完後那個成就感真的是很大，也很慶幸沒放棄這主題。