

中國文化大學
資訊工程學系
資訊系統專題

人臉辨識門禁系統

學 生：王 騰 緯

陳 柏 榮

邱 奕 融

許 家 豪

指導教授：李 志 仁

中 華 民 國 109 年 10 月

人臉辨識門禁系統

專題學生：王騰緯、陳柏榮、邱奕融、許家豪

指導教授：李志仁 博士

中國文化大學 資訊工程學系

摘要

鑰匙是人們最容易遺忘的東西，在外租屋或者是回自己家時，到達門口發現鑰匙不見又或者不在身上，那想必是非常不方便，隨著科技的演變與進步，我們得到了一種方法，人臉辨識，現在正處於此技術的重要開端，可用在許多探測、免現金支付、解鎖及提升安全性等，如果把人臉辨識技術結合門禁系統並使它普及的話，就可以解決前述的種種問題，不用帶著鑰匙，把辨識速度加快，就能非常方便，用一張臉就能開門，也同時可以觀看在你家門口出現的影像，解決開門之麻煩與家門安全性方面問題，本專題研發的人臉辨識門禁系統，利用電腦結合攝像頭和門鎖，也因應近期疫情影響，加裝熱像儀來測量體溫，完整的實現此次專題之成果，並希望能實際的良好運作。

關鍵詞：深度學習、卷積神經網路、預處理、人臉偵測、人臉辨識

指導教授_____ (簽名)

The Access Control System of Face Recongition

**Students: Teng-Wei Wang, Po-Jung Chen, Yi-Jung Chiu,
and Chia-Hao Hsu**

Advisor: Prof. Chih-Jen Lee

**Department of Computer Science and Information Engineering
Chinese Culture University**

ABSTRACT

Nowadays, the key is the most easily forgotten thing. When renting a house outside or returning to your home, you reach the door and find that the key is missing or forget to bring it, it must be very inconvenient. With the evolution and process of technology, we have obtained a method, face recognition, which is an important beginning of this technique now. It can be used in many places such as detections, cashless payments, unlocking, and improving security, etc. If the technology of face recognition is combined with the access control system and made it popular, it can solve the aforementioned problems. It is very convenient to accelerate the recognition without carrying a key. You can open the door just with one face and also can watch the image that appears at your doorway. It solves the trouble of opening door and the security. The access control system of the face recognition developed in this topic uses a computer combined with a camera and door lock. In response to the recent epidemic, we install a thermal imaging camera to measure temperature. And we completely want to achieve the result of this topic. At last, we hope that this system will be functional.

Keywords: Deep Learning, Convolutional Neural Network, Preprocess,
Face Detection, Face Recognition

目 錄

中文摘要	I
ABSTRACT	II
表 目 錄	V
圖 目 錄	VI
第 1 章 研究動機與目的	1
1.1 研究動機	1
1.2 研究目的	1
第 2 章 文獻探討	2
2.1 人臉特徵點偵測	2
2.2 卷積神經網路	3
2.2.1 卷積層	4
2.2.2 池化層	4
2.2.3 全鏈結層	5
2.3 機器學習:監督式學習	5
2.4 活體辨識	7
第 3 章 研究方法與軟硬體介紹	9
3.1 相關硬體	10
3.1.1 羅技 C310 HD 視訊攝像頭	10
3.1.2 Arduino	12
3.1.3 Raspberry Pi	15
3.1.4 FLiR Radiometric Lepton Dev Kit V2	18
3.2 相關軟體	20
3.2.1 Python	20
3.2.2 OpenCV	21
3.2.3 Tensorflow	22
第 4 章 整體實作	23

4.1 訓練人臉	23
4.2 人臉辨識相關流程.....	30
4.2.1 動態活體辨識.....	31
4.2.2 人臉辨識門禁系統.....	36
4.2.3 熱像儀	46
第 5 章 人力配置	47
第 6 章 結論	48
參考文獻	50

表 目 錄

表 3.1	羅技 C310 HD 攝像頭相關硬體參數	11
表 3.2	硬體 Arduino 相關硬體參數	13
表 3.3	硬體 Raspberry 相關硬體參數	16
表 3.4	硬體 FLiR Radiometric Lepton Dev Kit V2 相關硬體參數	19
表 5.1	參與本次專題組員與工作內容分配.....	47

圖 目 錄

圖 2.1	利用數字提取特徵示意圖[2].....	2
圖 2.2	特徵 68 點示意圖[3]	3
圖 2.3	卷積運算示意圖[5]	4
圖 2.4	池化層示意圖[6]	4
圖 2.5	全鏈結層示意圖[7]	5
圖 2.6	機器學習方式及各自演算法類別	5
圖 2.7	監督式學習示意圖[8]	6
圖 2.8	活體辨識成功示意圖[10]	7
圖 2.9	透過眼、嘴計算變化示意圖[12].....	8
圖 3.1	人臉辨識門禁系統架構圖	9
圖 3.2	訓練人臉部分軟硬體系統架構圖	9
圖 3.3	人臉辨識部分軟硬體系統架構圖	10
圖 3.4	羅技 C310 HD 視訊攝像頭[13]	11
圖 3.5	硬體 Arduino [14].....	13
圖 3.6	程式編輯畫面.....	14
圖 3.7	硬體 Raspberry Pi [15].....	15
圖 3.8	樹莓派開機後主頁面(Linux).....	17
圖 3.9	樹莓派操作畫面(Linux).....	17
圖 3.10	硬體 FLiR Radiometric Lepton Dev Kit V2 [16]	18
圖 3.11	程式語言 Python [17]	20
圖 3.12	軟體 OpenCV [18].....	21
圖 3.13	軟體 Tensorflow [19]	22
圖 4.1	人臉蒐集與訓練模型流程圖	23
圖 4.2	人臉特徵偵測 68 點實際測試圖	24
圖 4.3	擷取人臉	24
圖 4.4	擷取人臉程式碼.....	25

圖 4.5	抓取隨機亞洲人臉資料	26
圖 4.6	影像處理	26
圖 4.7	權重圖	27
圖 4.8	訓練圖	28
圖 4.9	準確率與損失率	28
圖 4.10	混淆矩陣	29
圖 4.11	人臉預測圖	29
圖 4.12	人臉辨識門禁系統流程圖	30
圖 4.13	辨識為圖片	31
圖 4.14	表情活體辨識.....	32
圖 4.15	臉部點位變化活體辨識	33
圖 4.16	使用隨機影片進行活體辨識	34
圖 4.17	使用刻意影片進行活體辨識	34
圖 4.18	活體辨識部分程式碼	35
圖 4.19	人臉辨識門禁系統 (上鎖).....	36
圖 4.20	人臉辨識門禁系統 (解鎖).....	37
圖 4.21	人臉辨識門禁系統 (正確人臉解鎖).....	38
圖 4.22	電腦畫面 (成功).....	39
圖 4.23	人臉辨識門禁系統 (陌生人解鎖).....	40
圖 4.24	電腦畫面 (失敗).....	41
圖 4.25	電腦畫面 (活體辨識失敗).....	42
圖 4.26	預測機率部分程式碼	43
圖 4.27	人臉辨識執行與解開陽極鎖部分程式碼	44
圖 4.28	門鎖相關 Arduino 程式碼	45
圖 4.29	熱像儀完整畫面	46

第 1 章 研究動機與目的

1.1 研究動機

現今生活中，人人都有可能忘記帶家門鑰匙的時候，隨著科技的發展，人臉辨識已經成為重要的一環，除了在手機上看到臉部辨識技術之外，如果使用臉部辨識作為的門禁系統，比起找鑰匙的時間快非常許多，此專題利用深度學習與特徵偵測，加上含有攝像頭的家庭門鎖，實現科技來改變生活上的便利，而最近也因為疫情關係，另外加裝了熱像儀攝像頭，利用此硬體測量準備要進入家中任何成員的體溫，更加實用。

1.2 研究目的

上述中，我們利用此技術，可以加快進門速度，拿一張臉就可以進入家門，這樣就不必擔心鑰匙要找、忘了帶或者不見的問題，另一方面，在安全性上，鑰匙被陌生人拿到隨便都能被複製，但人臉並沒有這麼簡單的，用機器訓練出各自的人臉資料，存入電腦並加以運算，並加強辨識的特徵，達到最好的效果，陌生人如果想要闖進門比起鑰匙是相當困難的，最終的目的，就是希望能把這樣的門禁系統讓越來越多人能使用，也因為加裝熱像儀，讓整體裝置更加的與眾不同，希望能在現今社會家庭中更加的普遍。

第 2 章 文獻探討

本章將會說明在此專題中所用到的技術並進一步探討，首先其注重人臉特徵部分該如何抓取，深度學習部分該如何使用，解釋何謂監督式學習，再解決不能以照片執行人臉辨識來預防不是真人的活體辨識。

2.1 人臉特徵點偵測

人臉特徵點偵測(Facial Landmark Detection)[1]是許多人臉分析方式中，最基本的組成部分，如判斷面部屬性、驗證人臉及人臉識別，儘管在這領域取得了足大的進步，在現在仍然是一個巨大的挑戰。

人臉特徵點偵測包括人臉特徵點定位、人臉對齊，特徵點(如嘴角、人中、眼睛等)為錨點，將人臉校正到同一個比較基準，取出特徵值來進行辨識。

而透過一些數字來表徵人臉訊息，這些數字就是我們要提取的特徵，常見的人臉特徵其一類是幾何特徵，幾何特徵是指眼睛、鼻子和嘴等面部特徵之間的幾何關係，如距離、面積和角度等。由於算法利用了一些直觀的特徵，計算量小，不過，其所需的特徵點不能精確選擇，限制了它的應用範圍。

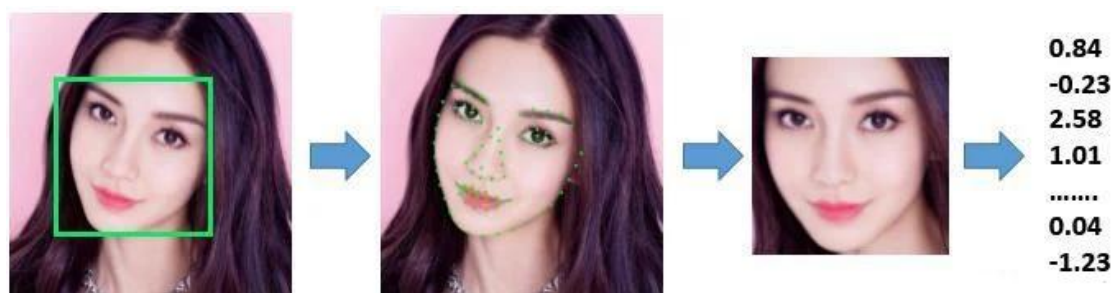


圖 2.1 利用數字提取特徵示意圖[2]

人臉特徵點檢測有還有下列等作用：

(1) 改善人臉識別偵測

通過人臉特徵點檢測將人臉對齊到平均臉，對齊後的人臉影像識別演算法更加有效。

(2) 人臉平均分配

利用人臉特徵點檢測的結果，將多個人臉進行融合，形成新的平均人臉。

(3) 人臉交換

利用人臉特徵點檢測的結果，對兩張人臉進行無縫換臉，將一張人臉換到另一張人臉上。

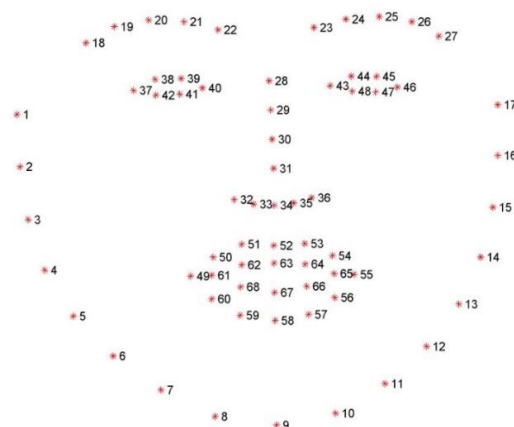


圖 2.2 特徵 68 點示意圖[3]

2.2 卷積神經網路

卷積神經網路(Convolutional Neural Network)[4]，常被稱為 CNN，是現在深度學習中，非常重要的一環，利用卷積運算方式，由點的比對轉成局部的比對得到較好的辨識結果，此結構近年用於圖像辨識以及語音分析已經相當普及，以下說明其中卷積層、池化層以及全鏈結層。

2.2.1 卷積層

譬如一張影像，把影像的每一點取為中心，取中心點周圍格子的點構成一個面，每一格給予不同的權重，並且計算加權總合，變成是這一點的輸出，在移動至下一點也一樣用相同方式處理，計算至最後一點為止。

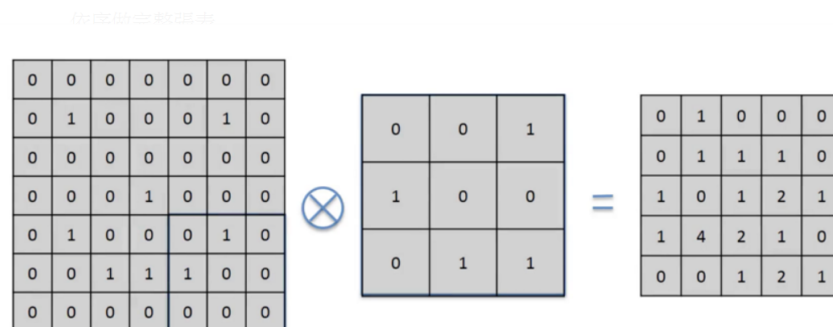


圖 2.3 卷積運算示意圖[5]

2.2.2 池化層

此作用是壓縮圖片並保留重要資訊的主要方式，池化層方法主要為 Max Pooling，會在圖片中取得不同的矩陣，並在矩陣中取得最大值。

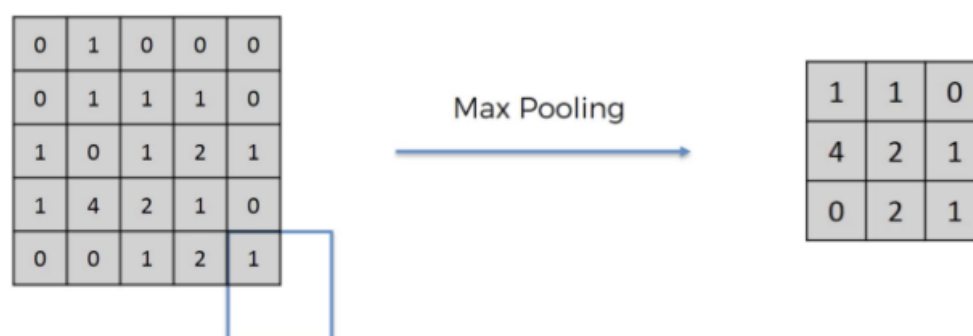


圖 2.4 池化層示意圖[6]

2.2.3 全鏈結層

簡單來說此層作用是先把前面的結果經過平坦層之後接到神經網路，前面的層學出了很多的特徵，而這些特徵的組合可以幫助我們分辨現在這個影像輸入是哪一個結果，是一個非常簡便的方法來把這些特徵實現分類的效果。

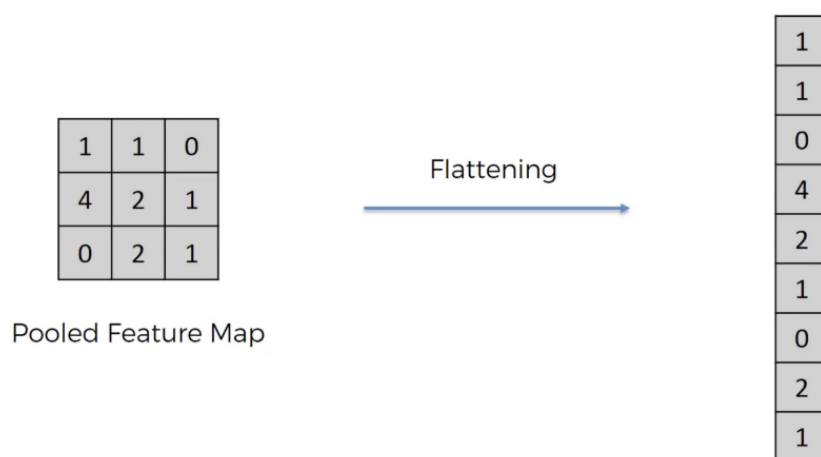


圖 2.5 全鏈結層示意圖[7]

2.3 機器學習:監督式學習

依訓練資料、產出判別的過程與結果不同，機器學習大致上可以分為三類：監督式學習 (Supervised Learning)、非監督式學習 (Unsupervised Learning) 與增強式學習 (Reinforcement Learning)，而其中我們選用監督式學習來做使用，並且進一步說明。

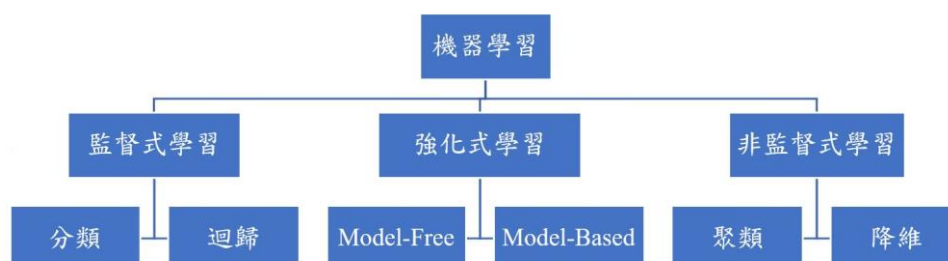


圖 2.6 機器學習方式及各自演算法類別

監督式學習(Supervised learning)也可以說成是一對一比照資訊的概念，電腦從標籤化 (labeled) 的資訊中分析模式後做出預測的學習方式。標記過的資料就好比標準答案，電腦在學習的過程透過對比誤差，一邊修正去達到更精準的預測，這樣的方式讓監督式學習有準確率高的優點。

監督式學習在分類問題和迴歸問題這兩大領域可以派上用場。分類問題要求演算法預測離散值，將輸入資料視為特定分類或分組的成員。例如，動物圖片的訓練資料集裡，代表每張圖片都會預先標記為貓、無尾熊或烏龜，接著會看演算法正確對其它無尾熊及烏龜的新圖片進行分類的精準度，對演算法進行評估。

另一方面，迴歸問題著眼於連續性資料。像是大家都很熟悉代數裡的線性迴歸：給予一個特定的 x 值，變數 y 的期望值會是多少。

一個更實際的機器學習例子與許多變數有關，像是按照平方英尺、地點和是否靠近公共交通站，預測舊金山某間公寓價格。

監督式學習方式需要倚靠大量的事前人工作業，將所有可能的特質標記起來，這過程相當繁複。當範圍擴大、資訊量增加，會更難去對資料標記出所有特徵，所以在面對未知領域時，幾乎是完全無法運作。

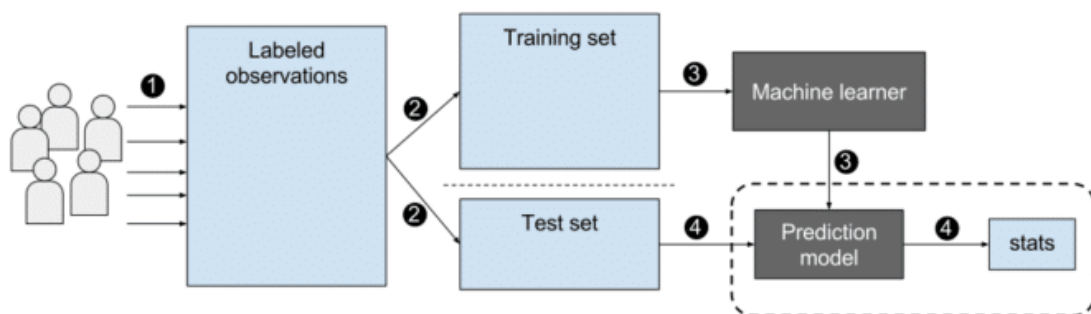


圖 2.7 監督式學習示意圖[8]

2.4 活體辨識

隨著人臉識別技術日趨成熟，然而人臉極易用照片、影片等方式進行複製，因此對合法用戶人臉的假冒是人臉識別安全的重要威脅，考慮到一但使用假人臉辨識成功，有可能對用戶造成重大的損失，因此需要為現有的人臉辨識系統開發可靠、高效的人臉活體辨識方法[9]。

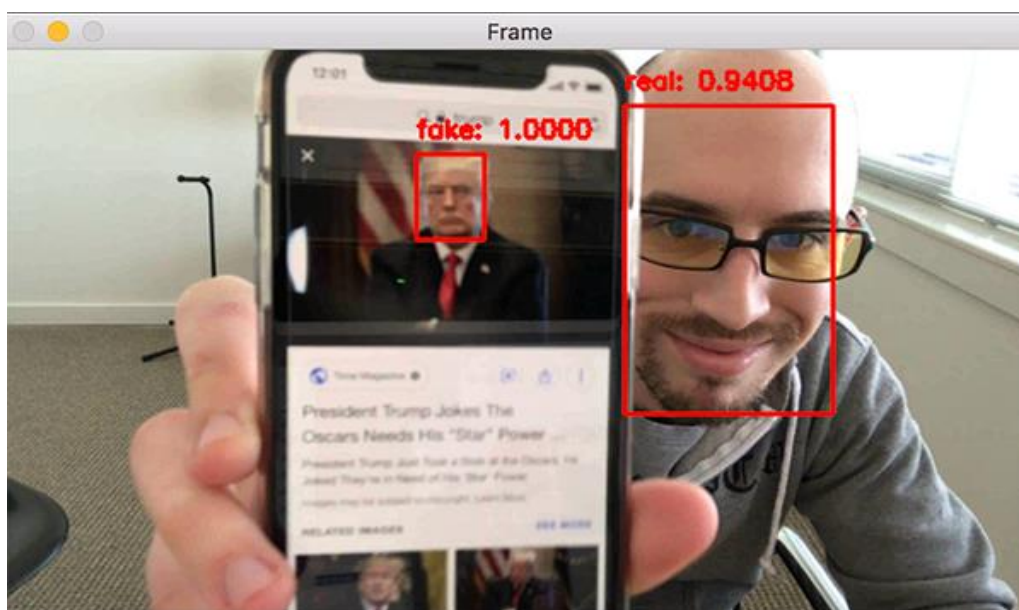


圖 2.8 活體辨識成功示意圖[10]

其中我們使用了最簡單的臉部表情動作活體辨識，利用眼和嘴的變化來防範照片[11]、影片和面具等辨識或非正常行為來准許通過辨識，透過這種隨機動作的人臉活體辨識技術，解決許多存在的漏洞問題。



圖 2.9 透過眼、嘴計算變化示意圖[12]

第 3 章 研究方法與軟硬體介紹

本章將會說明本此專題的系統架構(如圖 3.1)之外，還會接續介紹使用到的相關軟體、硬體，而也會在下一章呈現整個門禁系統的實作過程，系統整體分成兩部分，訓練人臉與人臉辨識。

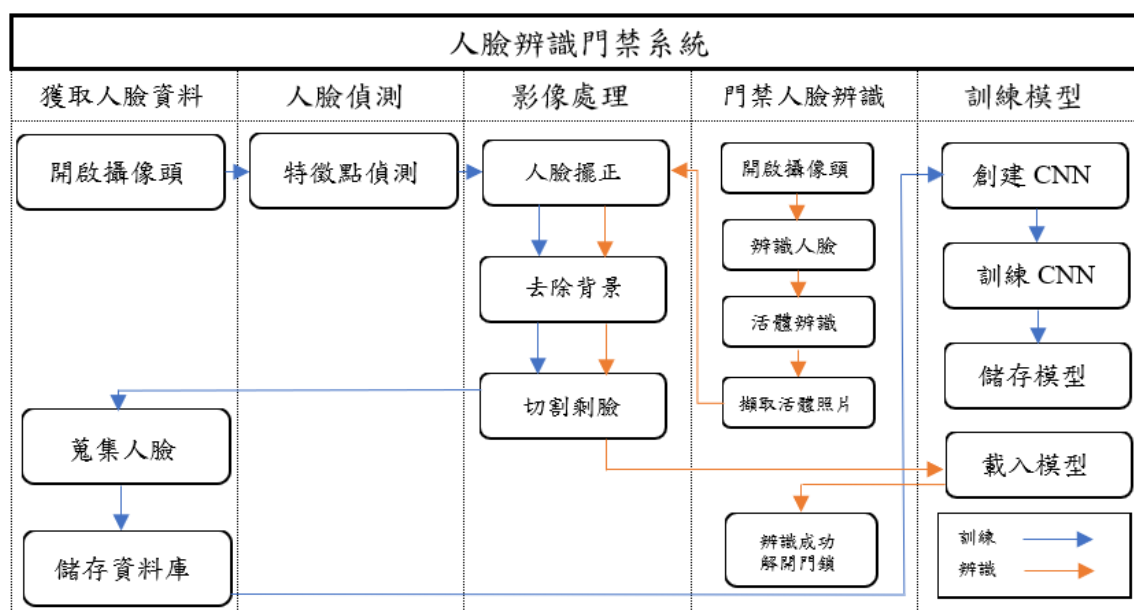


圖 3.1 人臉辨識門禁系統架構圖

首先開啟電腦視訊攝像頭偵測人臉擷取特徵後拍下照片，儲存至電腦本機資料庫，並建立模型 CNN 以及訓練，最後儲存模型。

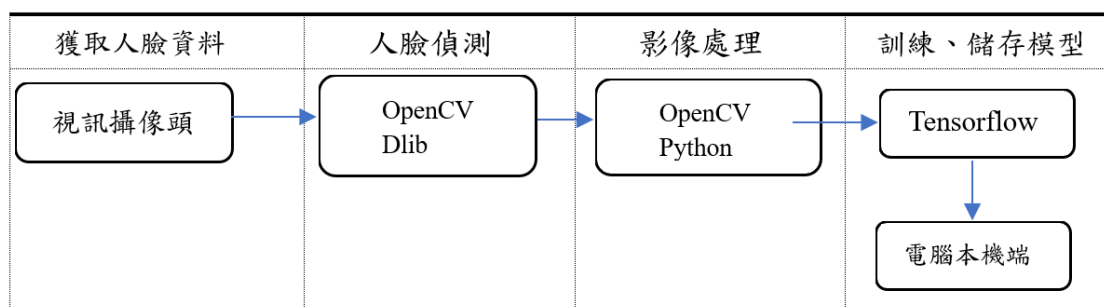


圖 3.2 訓練人臉部分軟硬體系統架構圖

第二部分開啟電腦視訊攝像頭，執行人臉辨識接著進行活體辨識，如果是真人拍下當時狀態，並把照片進行預處理，載入模型和資料庫進行比對，辨識成功後開啟磁扣門鎖。

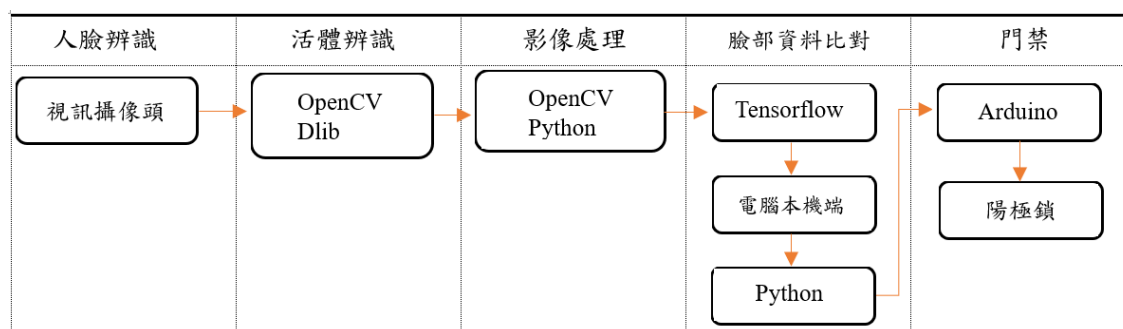


圖 3.3 人臉辨識部分軟硬體系統架構圖

在以上敘述完後，我們將介紹軟硬體與其中硬體參數。

3.1 相關硬體

3.1.1 羅技 C310 HD 視訊攝像頭

羅技 C310 HD 視訊攝像頭主要用於拍攝照片，最高可拍攝到高達 500 萬畫素的高解析度快照，也可輕鬆擷取 HD 視訊(720p)並上傳至 Facebook 或 Youtube，即使在微弱光源或不佳的背光下，具有獨特技術能智慧調整亮度，盡可能的把畫面調整至最好，含有內建式麥克風，如果想要配合電腦進行視訊線上對話也是沒有問題的。



圖 3.4 羅技 C310 HD 視訊攝像頭[13]

表 3.1 羅技 C310 HD 攝像頭相關硬體參數

相關規格	規格參數
影像擷取解析度	最高可達 1280x720 像素(720p/30fps)
照片	最高可達 500 萬畫素
連接介面	高速 USB2.0 認證
尺寸	31.3 公厘 x 71.2 公厘 x 68.4 公厘
重量	71.2 公克
電腦系統需求	Windows10、Windows8、Windows7
內建麥克風	單聲道
視野角度	60°

3.1.2 Arduino

Arduino 是一塊基於開放原始碼發展出來的 I/O 介面控制板，並且具有使用 類似 java、C 語言的開發環境，讓使用者可以快速使用 Arduino 語言與 Flash 或 Processing...等軟體，作出互動作品。透過 Arduino 很快的學習電子和感測器的基本知識，快速的設計、製作作品的原型，很容易與目前設計系所學的 FLASH、Virtool 等軟體整合，使得虛擬與現實的互動更加容易。互動的內容設計才是設計師的主要訴求，至於怎麼拼湊一個單晶片開發板，或 是當中涉及如何構築電路之類的知識，就並非設計師需要了解的，因此非常適合不具電子背景的人使用，以設計出各種不同的互動裝置。

Arduino 包含了硬體與軟體兩大部分，硬體部分是一個約手掌大小的控制板（寬 70mm X 高 54mm），核心使用八位元 ATMEGA8 系列的微控制器，提供 14 個數位式輸出/入端，6 個類比式輸出/入端，支援 USB 資料傳輸，可以使用自備電源(5V~9V)或是直接使用 USB 電源，使用者可以在數位式輸出/入端上接上不同的電子裝置，例如 LED 燈、喇叭、馬達，然後再由控制器來驅動燈的亮滅、喇叭發聲、馬達運轉。Arduino 控制板採用開放式源始碼設計的概念，電路設計圖、韌體都可以在網路上下載，稍具電子知識就可以自行製作，也可以在網路上用很便宜的價錢買到。

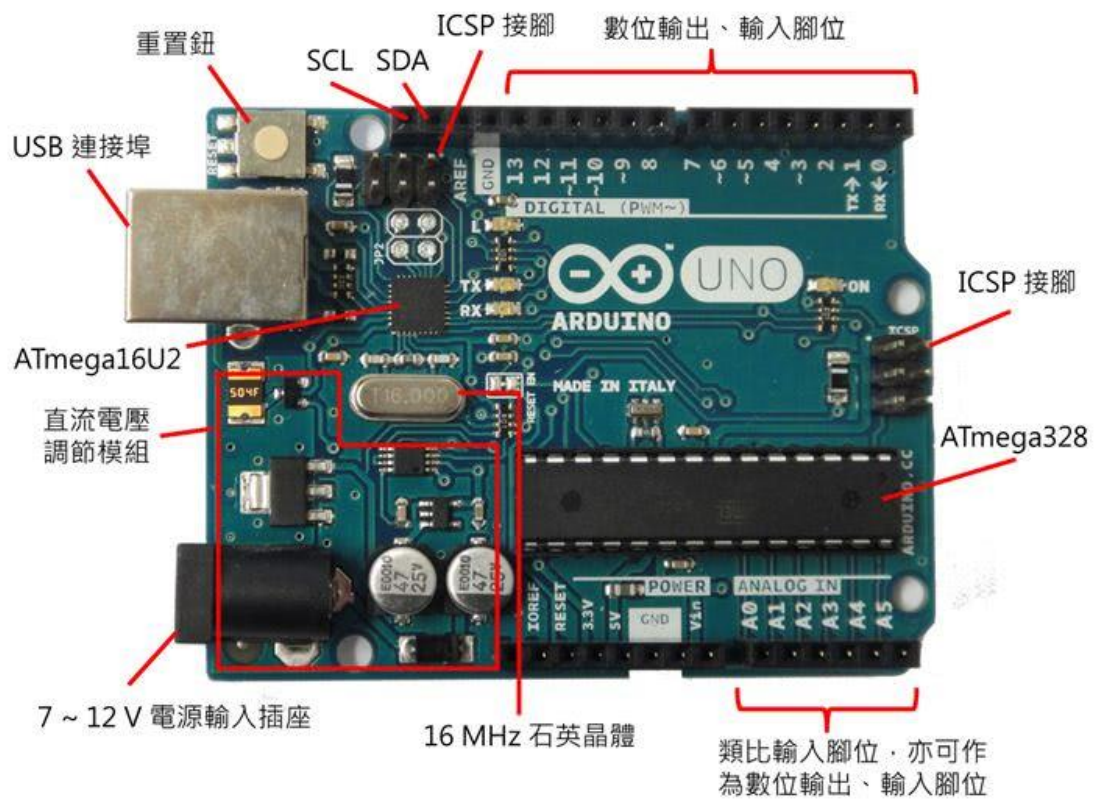


圖 3.5 硬體 Arduino [14]

表 3.2 硬體 Arduino 相關硬體參數

名稱	規格
微控制器	ATmega328P
工作電壓	5V
建議輸入電壓	7-12V
限制輸入電壓	6-20V

✓

→

📄

⬆

⬇

RowColumnScanning

```

void setup() {
    // initialize the I/O pins as outputs iterate over the pins:
    for (int thisPin = 0; thisPin < 8; thisPin++) {
        // initialize the output pins:
        pinMode(col[thisPin], OUTPUT);
        pinMode(row[thisPin], OUTPUT);
        // take the col pins (i.e. the cathodes) high to ensure that the LEDS are off:
        digitalWrite(col[thisPin], HIGH);
    }

    // initialize the pixel matrix:
    for (int x = 0; x < 8; x++) {
        for (int y = 0; y < 8; y++) {
            pixels[x][y] = HIGH;
        }
    }
}

void loop() {
    // read input:
    readSensors();

    // draw the screen:
    refreshScreen();
}

void readSensors() {
    // turn off the last position:
    pixels[x][y] = HIGH;
    // read the sensors for X and Y values:
    x = 7 - map(analogRead(A0), 0, 1023, 0, 7);
    y = map(analogRead(A1), 0, 1023, 0, 7);
    // set the new pixel position low so that the LED will turn on in the next
    // screen refresh:
    pixels[x][y] = LOW;
}

void refreshScreen() {
    // iterate over the rows (anodes):
    for (int thisRow = 0; thisRow < 8; thisRow++) {
        // take the row pin (anode) high:
        digitalWrite(row[thisRow], HIGH);
        // iterate over the cols (cathodes):
        for (int thisCol = 0; thisCol < 8; thisCol++) {
            // get the state of the current pixel;

```

圖 3.6 程式編輯畫面

3.1.3 Raspberry Pi

Raspberry Pi 是源自一個開放源程式碼的硬體專案平台,該平台包括一塊具備簡單 I/O 功能的電路板以及一大堆的 Linux 軟體。Raspberry Pi 可以用來開發交互產品,比如它可以讀取大量的開關和感測器信號,並且可以控制電燈、電機和其他各式各樣的物理設備;Raspberry Pi 也可以開發出與 PC 一樣的周邊裝置,也可以運行在 Linux PC 上的軟體進行通信。Raspberry Pi 的硬體電路板可以自行焊接組裝,也可以購買已經組裝好的模組。



圖 3.7 硬體 Raspberry Pi [15]

表 3.3 硬體 Raspberry 相關硬體參數

名稱	規格
作業系統	Liunx
使用電力	3.5W
CPU 處理器	ARM1176JZF-S 700 MHz
儲存	SD 卡
Memory	512MByte
輸出顯示	HDMI、RCA
輸入連結	USB
電源	透過 Micro USB
聲音輸出	Audio Jack



圖 3.8 樹莓派開機後主頁面(Linux)

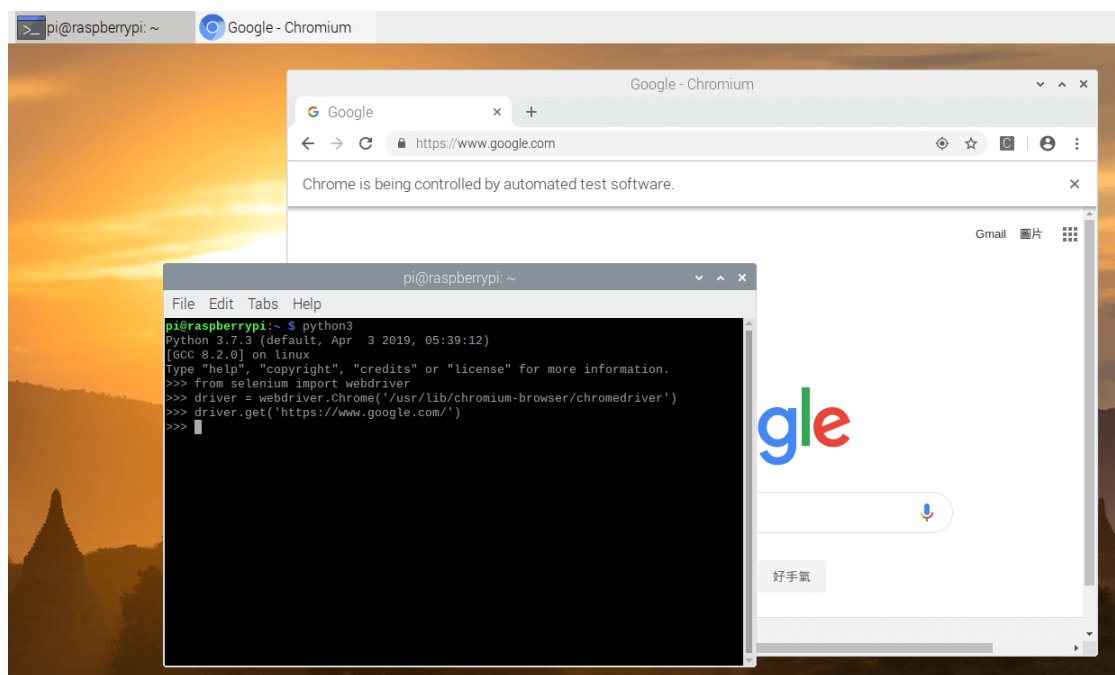


圖 3.9 樹莓派操作畫面(Linux)

3.1.4 FLiR Radiometric Lepton Dev Kit V2

FLiR Radiometric Lepton Dev Kit V2 紅外線熱影像熱成像開發套件，基於 Lepton 2.5 長波紅外（LWIR）成像儀。使用此套件，專門針對 Arduino、Raspberry Pi 或任何基於 ARM 的開發平台，能夠將 FLiR 的熱成像技術導入專案，FLiR Dev Kit 紅外線熱影像開發套件無須複雜的接線，您可以輕鬆透過杜邦線與麵包板即可開始進行開發與設計

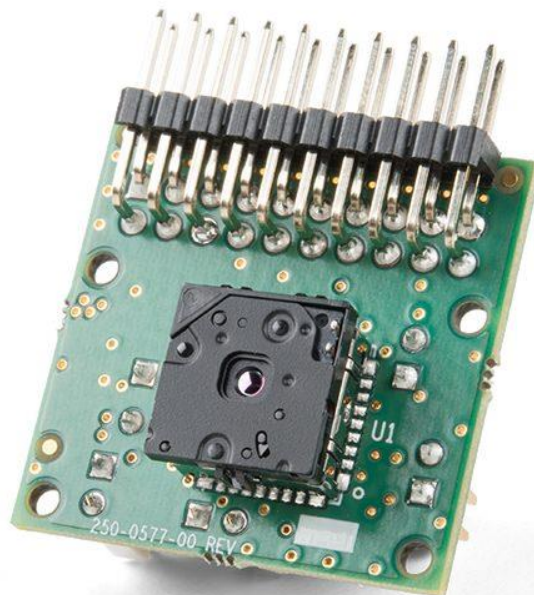


圖 3.10 硬體 FLiR Radiometric Lepton Dev Kit V2 [16]

表 3.4 硬體 FLiR Radiometric Lepton Dev Kit V2 相關硬體參數

名稱	規格
LWIR 感測器	可偵測波長 8 到 14 μm
視角	水平 50 度，對角 60 度
像素	80 (h) \times 60 (v)
熱靈敏度	小於 50 mK
相機模組介面	提供 SPI 及 I2C
影像響應時間	小於 1.2 秒
功耗	標準值 150 mW
輸入電壓	3V ~ 5.5V

3.2 相關軟體

3.2.1 Python

Python 是物件導向程式及高階程式語言，也是直譯式程式語言，Python 強調對程式語言的語句易讀、易懂、易學(簡潔和清晰的語法)特點及加快程式開發的時效，方便使用，可以完成各種難度的應用，並可在大多數系統中運行，以減少開發及維護成本的觀念進行發展。

Python 同時支援 modules(模組)和 packages(包裝)，另外 Python 為跨平台程式語言也支援 unicode 字元，功能強大而完善的通用型語言，可以用於很多種軟體開發動態程式，使得 Python 非常有吸引力，發展至今已有十多年的歷史，成熟且穩定。

通常，有許多程式設計者接觸 Python 後都會被她的優點所吸引，並提供了許多自行開發的 library(函式庫)以提供其他 Python 程式設計者下載使用，所以具有豐富和龐大的類別函式庫，可以負荷支持平常大部份的應用，使得現今 Python 發展迅速及使用廣泛，相關資源也相當豐富。



圖 3.11 程式語言 Python [17]

3.2.2 OpenCV

OpenCV 全名是 Open Source Computer Vision Library，是當今最知名、也最被廣泛採用的影像處理函式庫，它是由 Intel 發起並參與開發，以 BSD 授權條款發行，可在商業和研究領域中免費使用。

OpenCV 由 C 和 C++ 函式庫所構成，也提供了 Python、Ruby、MATLAB 等語言的介面，實現了影像處理和計算機視覺方面的很多通用演算法。

OpenCV 在影像處理方面應用廣泛，可以讀取儲存圖片、視訊、矩陣運算、統計等，可用在物體追蹤、人臉辨識、紋理分析、動態視訊的影像處理等。

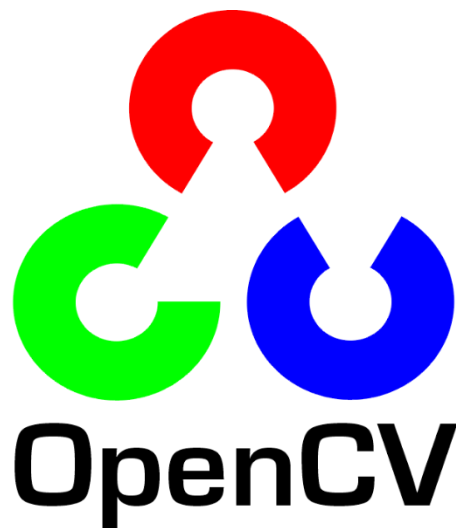


圖 3.12 軟體 OpenCV [18]

3.2.3 Tensorflow

Tensorflow 最初為 Google Brian 所開發。在 2015 時，Google 將之開源，為現今重要的深度學習框架之一，它支援各式不同的深度學習演算法，並已應用於各大企業服務上，Ex: Google、Youtube、Airbnb、Paypal ...等。此外，Tensorflow 也支援在各式不同的設備上運行深度學習 Ex: Tensorflow Lite、Tensorflow.js 等等。

TensorFlow 可使用 Python 與 C++運行，但較多開發者使用 Python，有 CPU 與 GPU 兩個版本，訓練大型的網路時，一般會選用 GPU 的版本訓練，速度約快數十倍。

Tensorflow 為目前最受歡迎的機器學習、深度學習開源專案，因此，在 AI 時代下，Tensorflow 是一個值得投資及好好學習的深度學習框架。



圖 3.13 軟體 Tensorflow [19]

第 4 章 整體實作

本章將敘述實作人臉辨識門禁系統相關環節，首先講述我們是如何訓練人臉資料的流程，再介紹我們如何實現整體人臉辨識門禁系統的流程。

4.1 訓練人臉

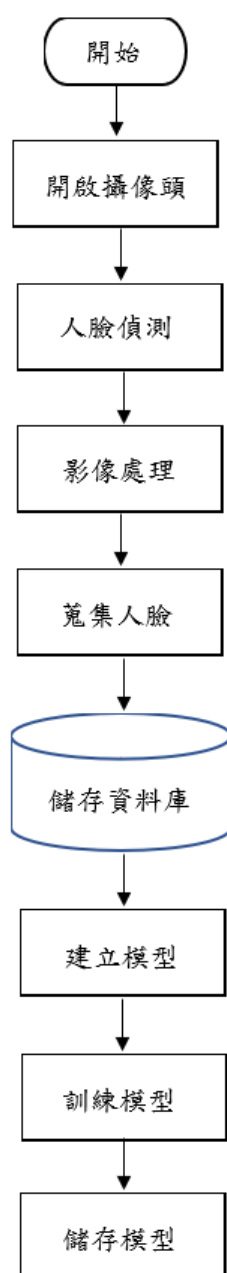


圖 4.1 人臉蒐集與訓練模型流程圖

首先，把四個人臉資料及放進電腦中，開啟視訊攝像頭，使用 OpenCV 並結合 Dlib 函式庫，利用人臉特徵點偵測，如偵測到人臉後，儲存全景照片各 1000 張。



圖 4.2 人臉特徵偵測 68 點實際測試圖



圖 4.3 擷取人臉

下圖 4.4 程式碼為利用 OpenCV 與 Dlib 函式庫來拍攝人臉，並把拍下的人臉存在本機端。

```
1 import dlib
2 import cv2
3 # 載入並初始化檢測器
4 detector = dlib.get_frontal_face_detector()
5 camera = cv2.VideoCapture(0)
6 if not camera.isOpened():
7     print("cannot open camera")
8     exit(0)
9 j=0
10 while True:
11     ret, frame = camera.read()
12     if not ret:
13         break
14     frame_new = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
15     # 檢測臉部
16     dets = detector(frame_new, 1)
17
18     # 查詢臉部位置
19     for i, face in enumerate(dets):
20
21         # 繪製臉部位置
22
23         # 儲存臉部圖片
24         img1=frame[face.top():face.bottom(),face.left():face.right()]
25
26         img_name = '%s/%d.jpg'%( 'C:/Users/vince/face/train/vincent',j)
27         cv2.imwrite(img_name,frame)
28         cv2.rectangle(frame, (face.left(), face.top()), (face.right(), face.bottom()),(0,255,0), 3)
29         cv2.imshow("Camera", frame)
30
31         j+=1
32
33         if (j>100):
34             break
35
36         key = cv2.waitKey(1)
37         if key == 27:
38             break
39 camera.release()
40 cv2.destroyAllWindows()
```

圖 4.4 擷取人臉程式碼

而為了解決陌生人問題，我們抓取 2000 張亞洲人臉設為其他人，並且同樣進行人臉特徵點偵測。



圖 4.5 抓取隨機亞洲人臉資料

完成後，進行影像處理讓訓練變得更好，把照片先擺正，去除背景並切至剩臉的部分，在此很明顯的看出，把不重要的許多干擾資料刪除，留下各個人臉的特徵有很大的幫助。

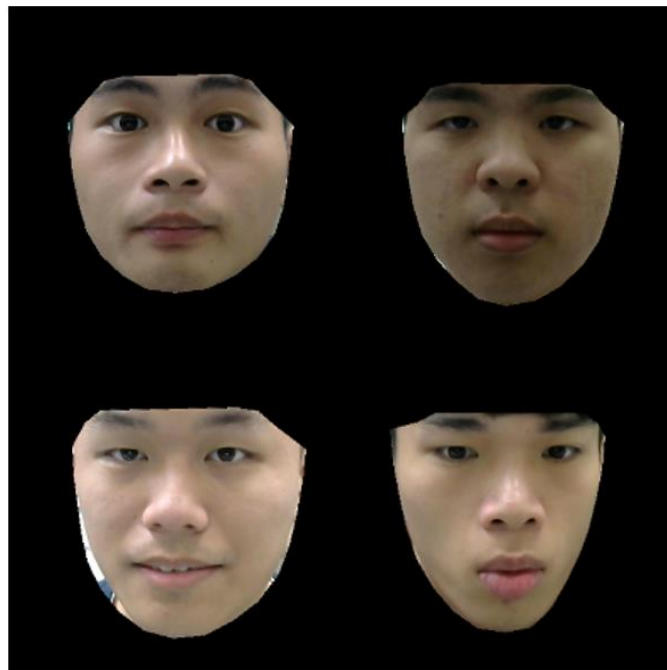


圖 4.6 影像處理

接著是訓練部分，下載這些切好的人臉資料當作 train data，其中分 20%給 test data，做完後接續創建模型 CNN，經過卷積層、池化層、全鏈結層和輸出後，顯示權重，訓練、混淆矩陣與訓練成功率損失率各圖。

Model: "sequential_2"

Layer (type)	Output Shape	Param #
=====		
conv2d_5 (Conv2D)	(None, 100, 100, 32)	896
max_pooling2d_5 (MaxPooling2)	(None, 50, 50, 32)	0
dropout_6 (Dropout)	(None, 50, 50, 32)	0
conv2d_6 (Conv2D)	(None, 50, 50, 32)	9248
max_pooling2d_6 (MaxPooling2)	(None, 25, 25, 32)	0
dropout_7 (Dropout)	(None, 25, 25, 32)	0
conv2d_7 (Conv2D)	(None, 25, 25, 32)	9248
max_pooling2d_7 (MaxPooling2)	(None, 12, 12, 32)	0
dropout_8 (Dropout)	(None, 12, 12, 32)	0
conv2d_8 (Conv2D)	(None, 12, 12, 64)	18496
max_pooling2d_8 (MaxPooling2)	(None, 6, 6, 64)	0
dropout_9 (Dropout)	(None, 6, 6, 64)	0
flatten_2 (Flatten)	(None, 2304)	0
dense_3 (Dense)	(None, 512)	1180160
dropout_10 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 5)	2565
=====		
Total params: 1,220,613		
Trainable params: 1,220,613		
Non-trainable params: 0		

圖 4.7 權重圖

```

Train on 6725 samples, validate on 1682 samples
Epoch 1/5
6725/6725 [=====] - 7s 976us/step - loss: 1.0572 - accuracy: 0.5768 - val_loss: 0.3136 - val_accuracy: 0.9608
Epoch 2/5
6725/6725 [=====] - 4s 609us/step - loss: 0.1176 - accuracy: 0.9665 - val_loss: 0.0155 - val_accuracy: 0.9976
Epoch 3/5
6725/6725 [=====] - 4s 611us/step - loss: 0.0325 - accuracy: 0.9929 - val_loss: 0.0114 - val_accuracy: 0.9952
Epoch 4/5
6725/6725 [=====] - 4s 611us/step - loss: 0.0161 - accuracy: 0.9960 - val_loss: 0.0254 - val_accuracy: 0.9917
Epoch 5/5
6725/6725 [=====] - 4s 614us/step - loss: 0.0119 - accuracy: 0.9969 - val_loss: 0.0235 - val_accuracy: 0.9917

```

圖 4.8 訓練圖

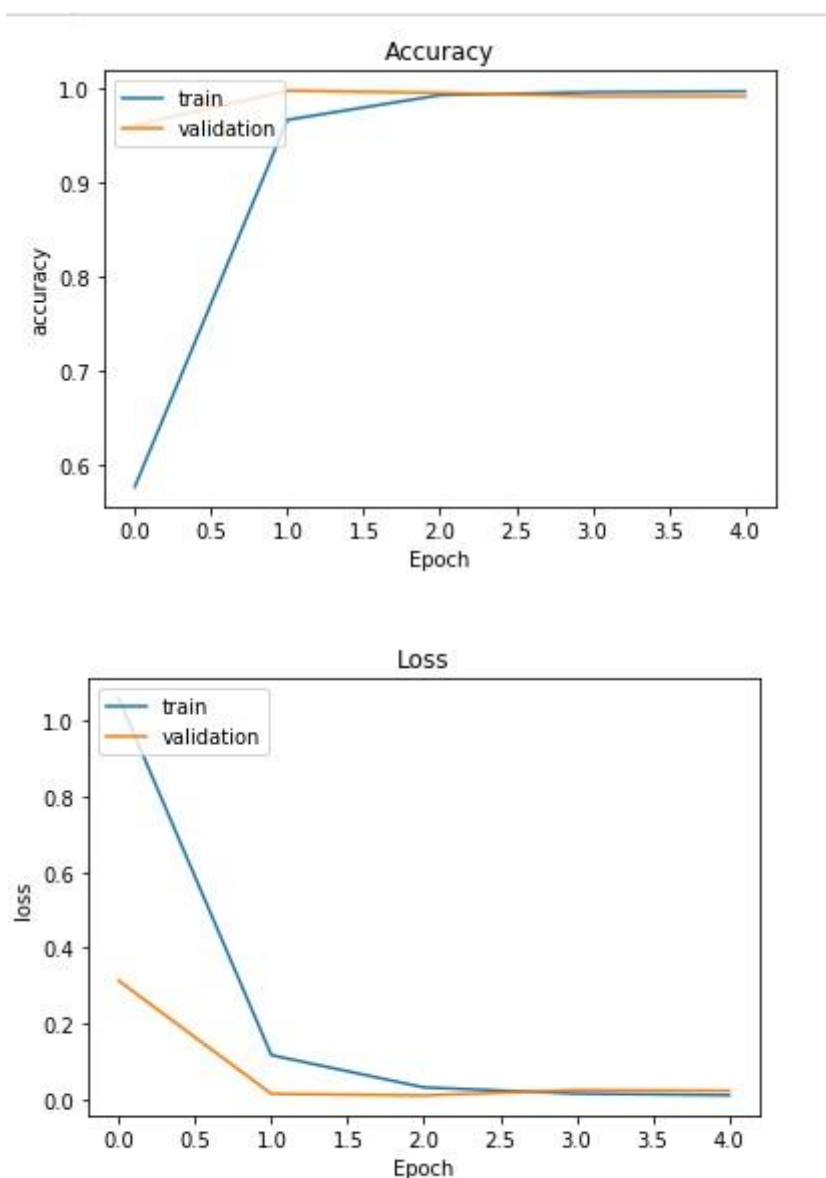


圖 4.9 準確率與損失率

Out[11]:

predict	0	1	2	3	4
label					
0	354	0	0	0	0
1	0	351	0	0	0
2	0	0	359	0	0
3	0	0	0	388	0
4	0	1	10	3	216

圖 4.10 混淆矩陣

而最後顯示五組人臉辨識的正確與錯誤，都是對應到對的人。

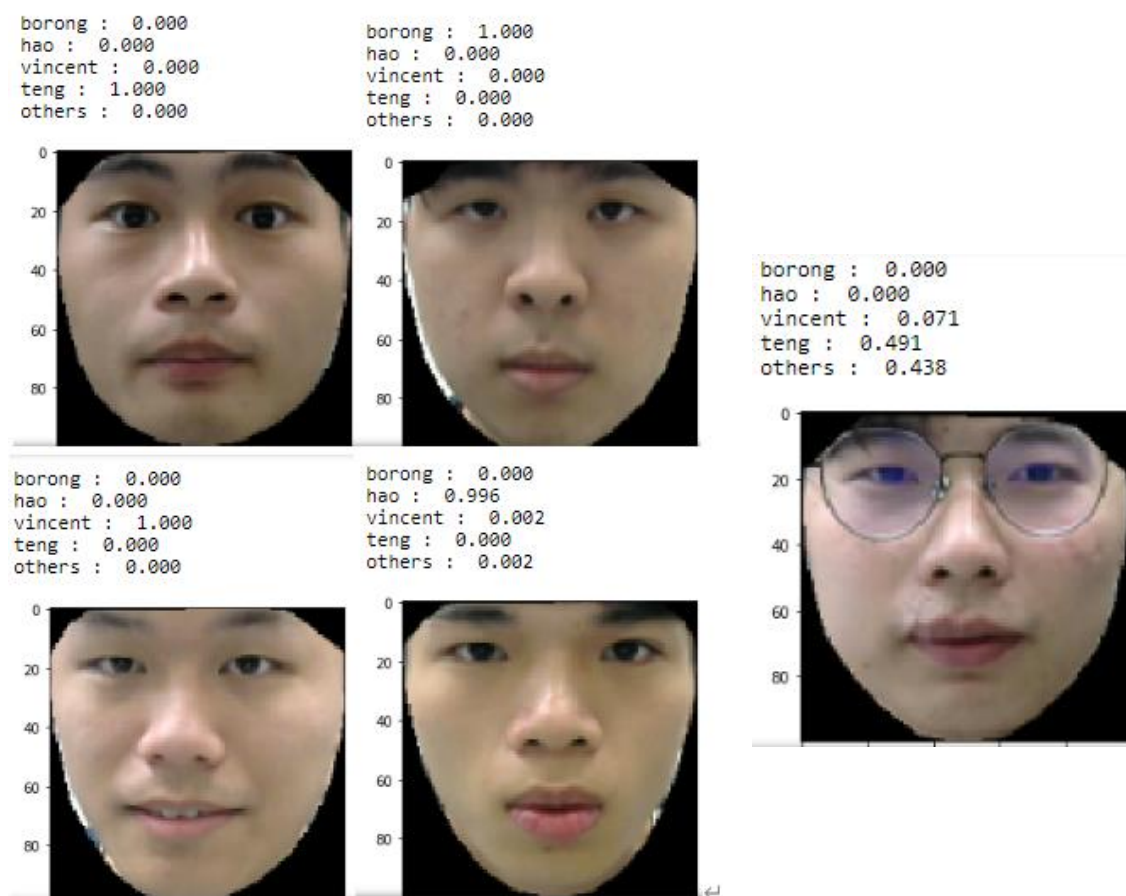


圖 4.11 人臉預測圖

4.2 人臉辨識相關流程

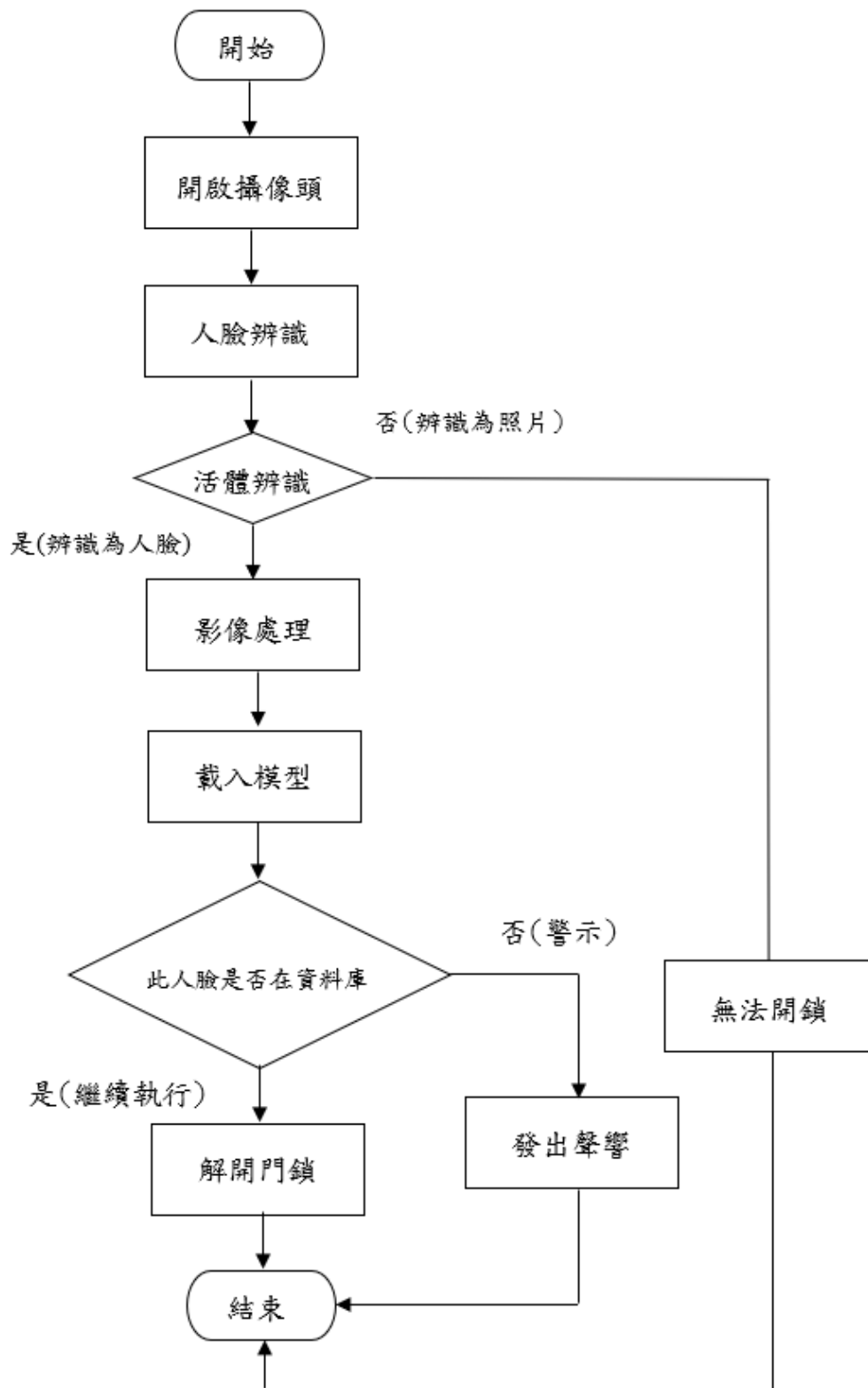


圖 4.12 人臉辨識門禁系統流程圖

4.2.1 動態活體辨識

第一次使用的方式已成功利用嘴、眼變化計算部位，實現動態表情來預防不為活體問題，整體辨識還並不是很自然，但初步達到想要的活體辨識效果，如果是真人顯示"people"，否則皆為顯示"picture"。



圖 4.13 辨識為圖片

下圖 4.14 想表現出變化在於表情的不同。

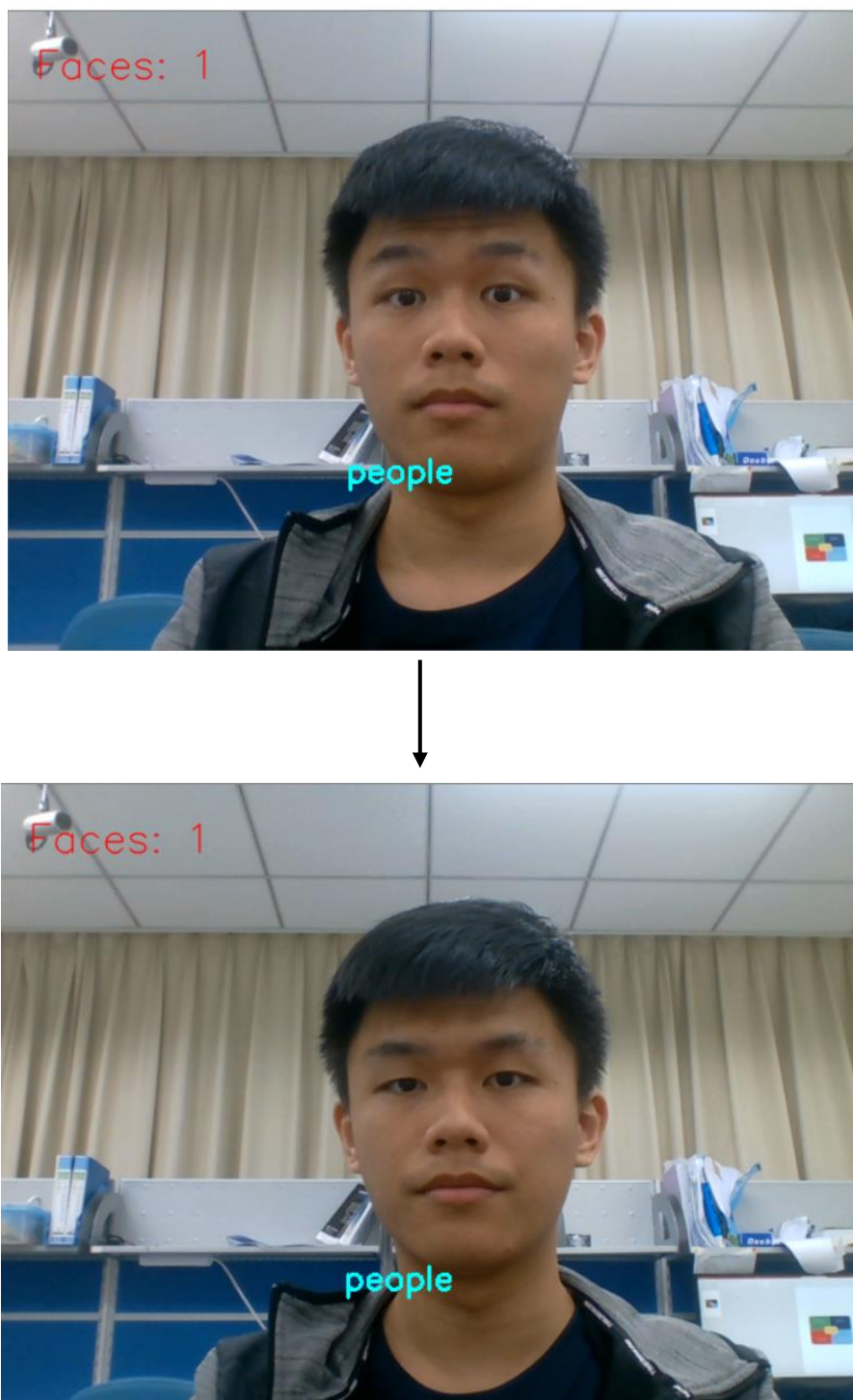


圖 4.14 表情活體辨識

使用前述的方法發現，除了不夠自然之外，如果表情變化不夠大有機會辨識不為真人，接下來第二種方式，利用臉部點位變化，讓頭左移右移讓人臉辨識更加自然活用。

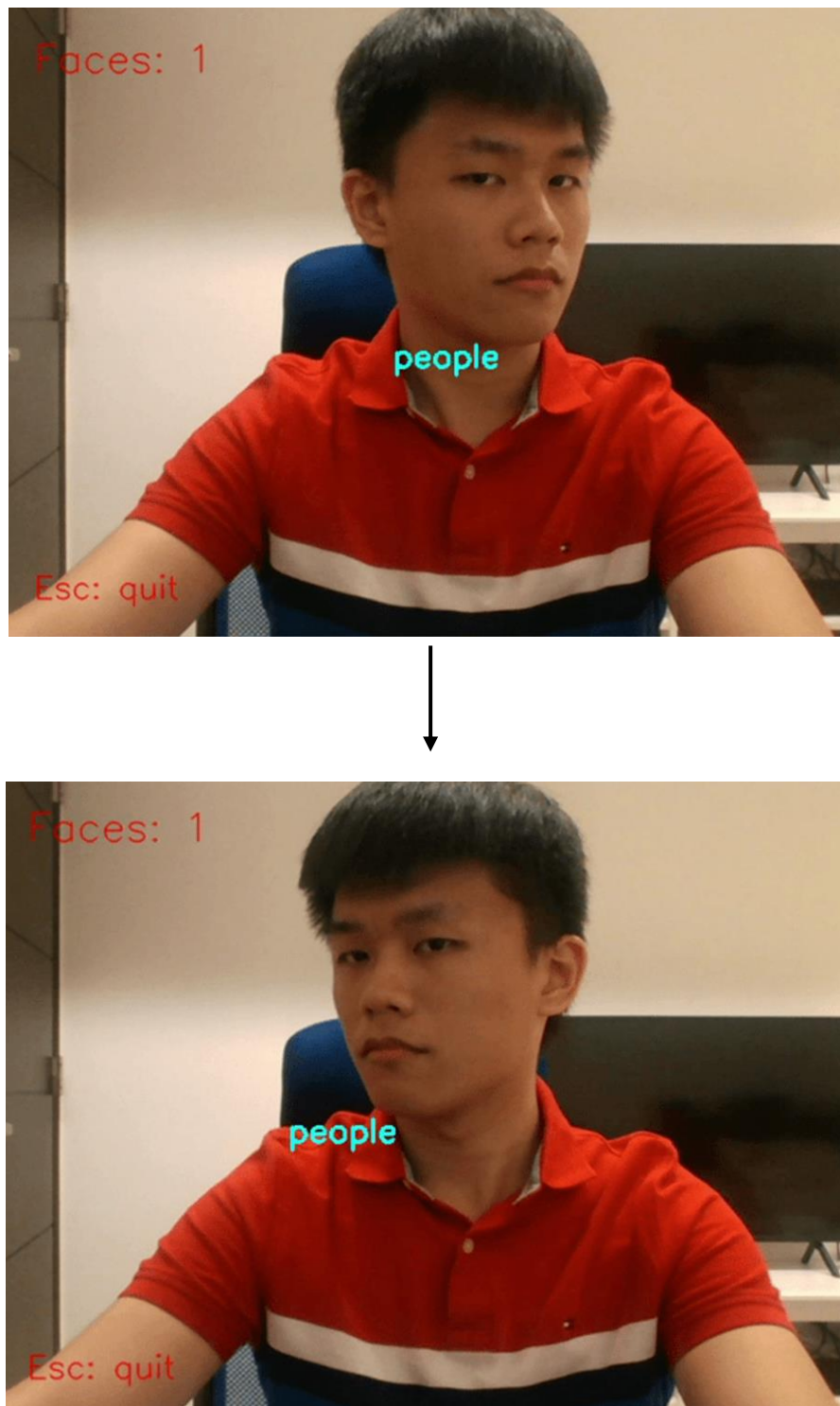


圖 4.15 臉部點位變化活體辨識

使用隨機找來的影片來辨識此門禁系統也成功判斷出結果。



圖 4.16 使用隨機影片進行活體辨識

但如果是利用手機以刻意拍攝臉部點位變化的影片，還是會成功解鎖的，雖然已經有做到相關技術，但還是沒有完美的解決辦法。



圖 4.17 使用刻意影片進行活體辨識

下圖 4.18 為利用 OpenCV 與 Dlib 函式庫中相關臉部參數來計算臉部點位，實現活體辨識。

```
def learning_face(self):
    face_width_max = 0
    face_width_min = 0
    count=0
    success=0
    while(self.cap.isOpened()):

        flag, im_rd = self.cap.read()

        k = cv2.waitKey(100)

        img_gray = cv2.cvtColor(im_rd, cv2.COLOR_RGB2GRAY)

        faces = self.detector(img_gray, 0)

        font = cv2.FONT_HERSHEY_SIMPLEX

        if(len(faces)!=0):

            for i in range(len(faces)):

                for k, d in enumerate(faces):

                    self.face_width = d.right() - d.left()

                    shape = self.predictor(im_rd, d)

                    face_width_new=(shape.part(15).x-shape.part(34).x)/(shape.part(28).y-shape.part(9).y)
                    face_width_right=(shape.part(15).x-shape.part(34).x)/(shape.part(28).y-shape.part(9).y)
                    face_width_left=(shape.part(34).x-shape.part(3).x)/(shape.part(28).y-shape.part(9).y)

                    if(face_width_max==0):
                        face_width_max = face_width_new
                        face_width_min = face_width_new
                    else:
                        if(face_width_new > face_width_max):
                            face_width_max = face_width_new

                        if(face_width_new < face_width_min):
                            face_width_min = face_width_new

                    if((face_width_max-face_width_min) > 0.3 and face_width_right-face_width_left < 0.07 and face_width_right-face_width_left > -0.07):
                        success = 1
                        break
                    count+=1
                    if(face_width_right-face_width_left < 0.07 and face_width_right-face_width_left > -0.07):
                        cv2.imwrite("C:/Users/as100/Downloads/20200528/prediction/0.png", im_rd)

                if(count==20 or success == 1):
                    print("")
                    break
            else:
                cv2.putText(im_rd, "No Face", (20, 50), font, 1, (0, 0, 255), 1, cv2.LINE_AA)

        im_rd = cv2.putText(im_rd, "Esc: quit", (20, 450), font, 0.8, (0, 0, 255), 1, cv2.LINE_AA)

        if(cv2.waitKey(10) == 27):
            break

        cv2.imshow("camera", im_rd)

    self.cap.release()

    cv2.destroyAllWindows()
    return success
```

圖 4.18 活體辨識部分程式碼

4.2.2 人臉辨識門禁系統

此小節將先介紹完整人臉辨識門禁系統實體模樣，再進一步介紹實際操作流程。

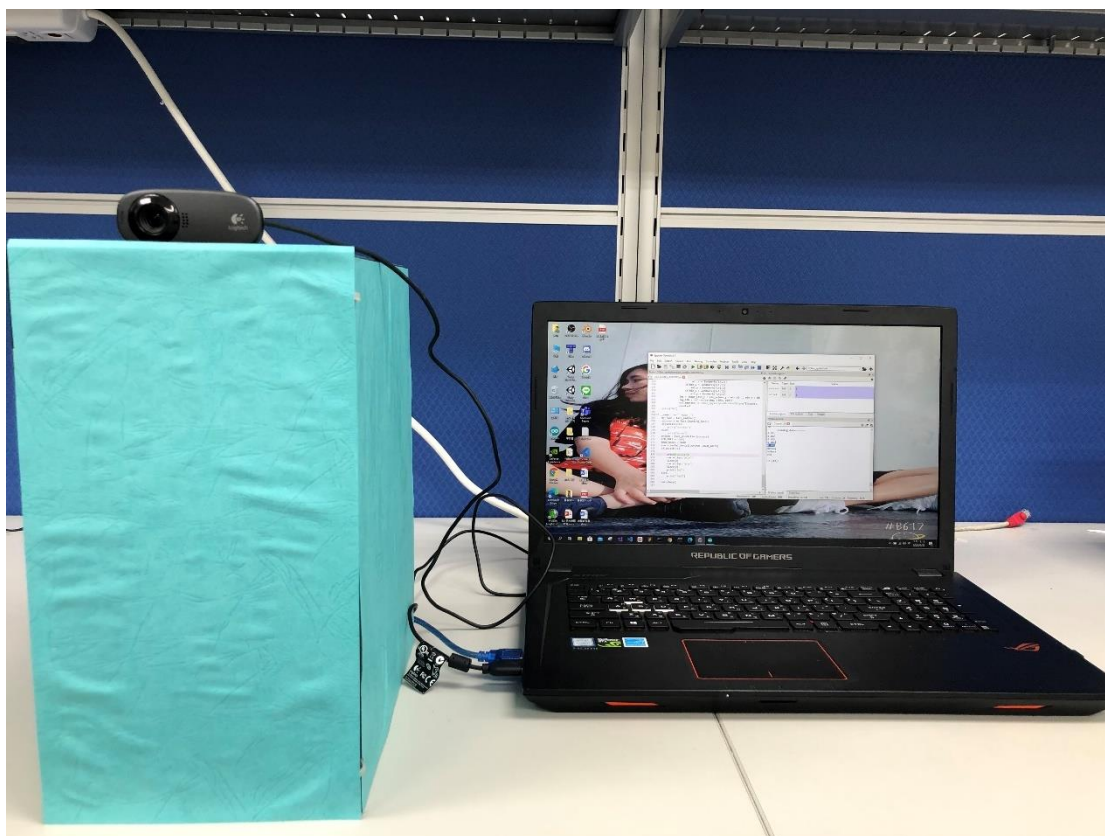


圖 4.19 人臉辨識門禁系統 (上鎖)

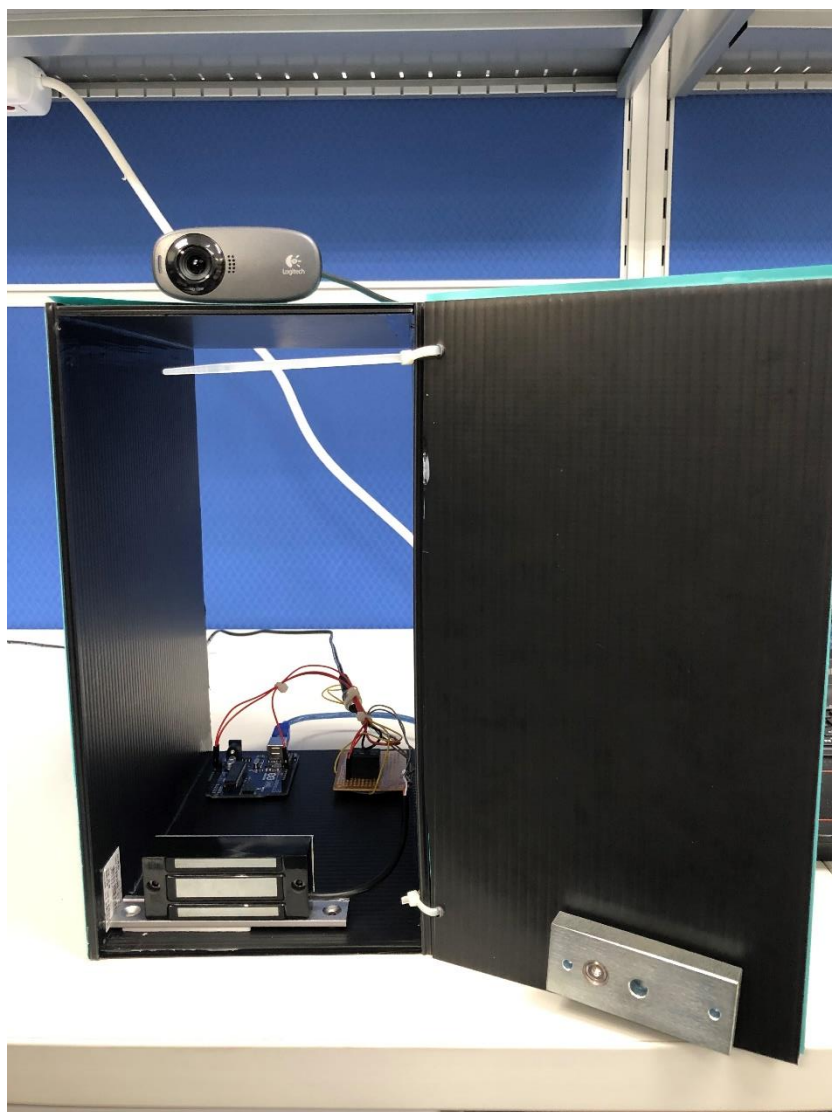


圖 4.20 人臉辨識門禁系統 (解鎖)

首先是人臉資料已在資料庫裡的正確示範，開啟攝像頭，對著它進行活體辨識偵測，過關後拍下一張照片，經過影像處理，然後載入模型比對是否為資料裡的人臉，正確後解鎖人臉。

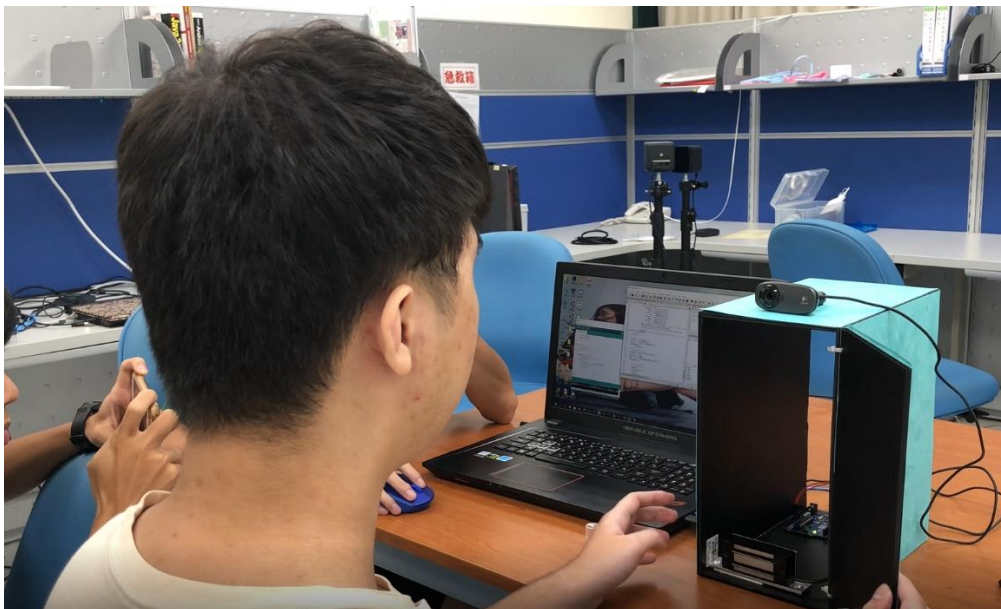
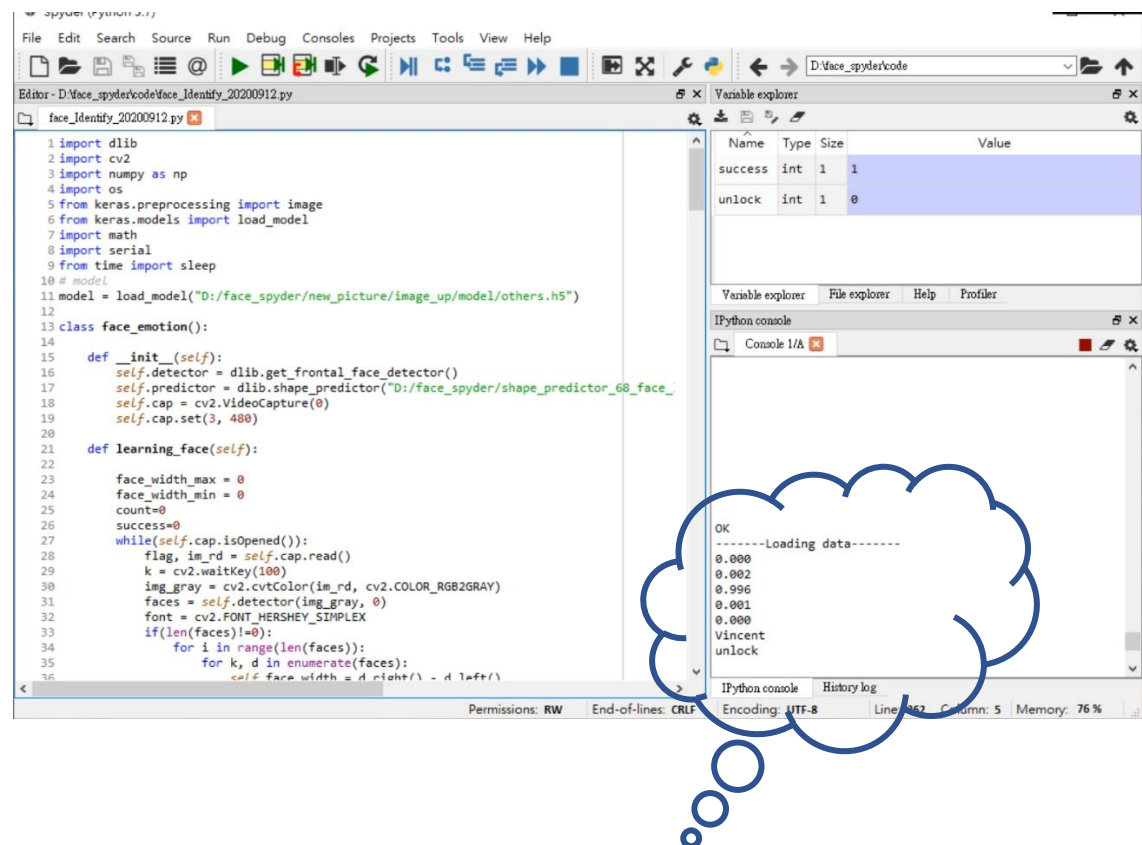


圖 4.21 人臉辨識門禁系統 (正確人臉解鎖)

下圖 4.22 為解鎖時電腦畫面，先顯示 OK 表示活體辨識過關，接著表示此刻人臉比對資料庫與誰最相近機率，再顯示解鎖此門禁系統的人為誰，最後則解開陽極鎖。



```
OK
-----Loading data-----
0.000
0.002
0.996
0.001
0.000
Vincent
unlock
```

圖 4.22 電腦畫面 (成功)

接下來，為當陌生人解鎖此人臉辨識門禁系統的情況下，活體辨識依然會過關，但最後並沒有成功解開陽極鎖。

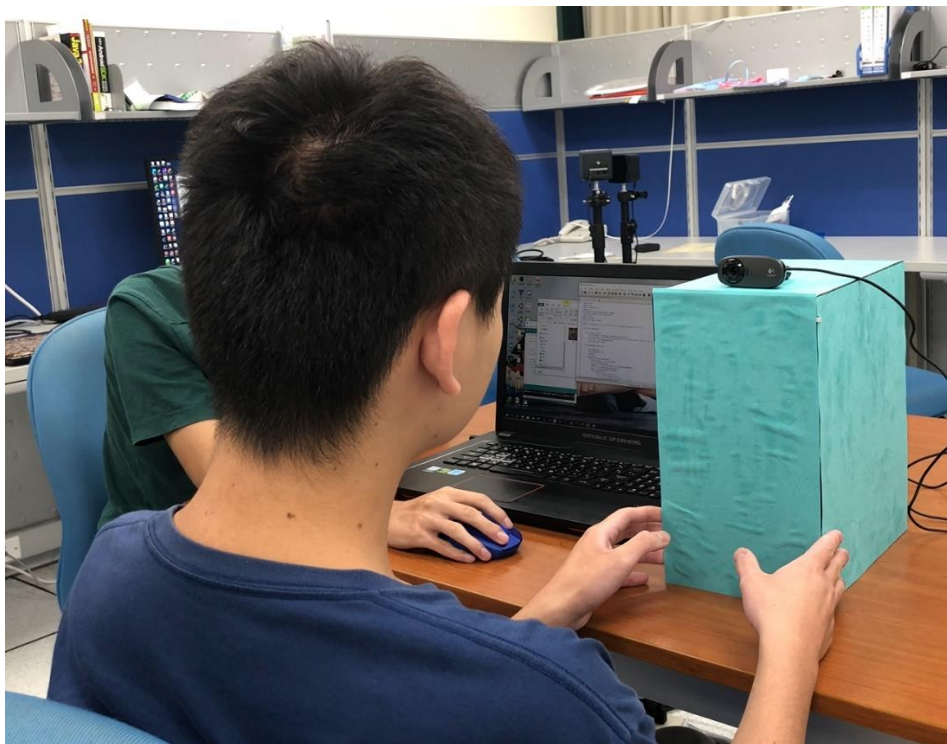


圖 4.23 人臉辨識門禁系統 (陌生人解鎖)

而如下圖 4.24 一樣成功過關活體辨識後，進行人臉比對，結果出來為陌生人機率最高，因此顯示解鎖此人臉為 other。

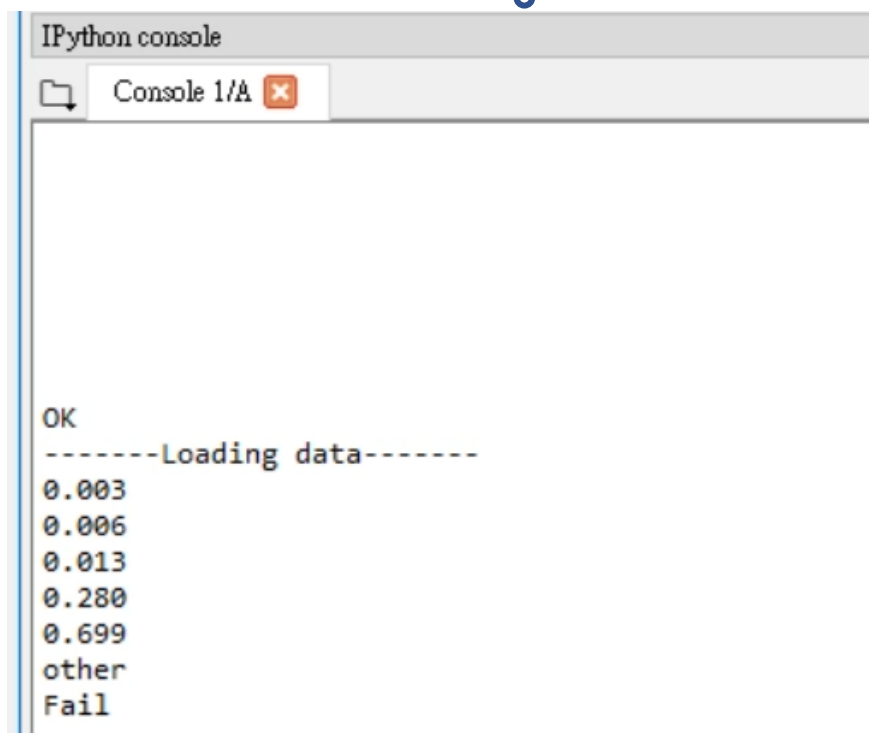
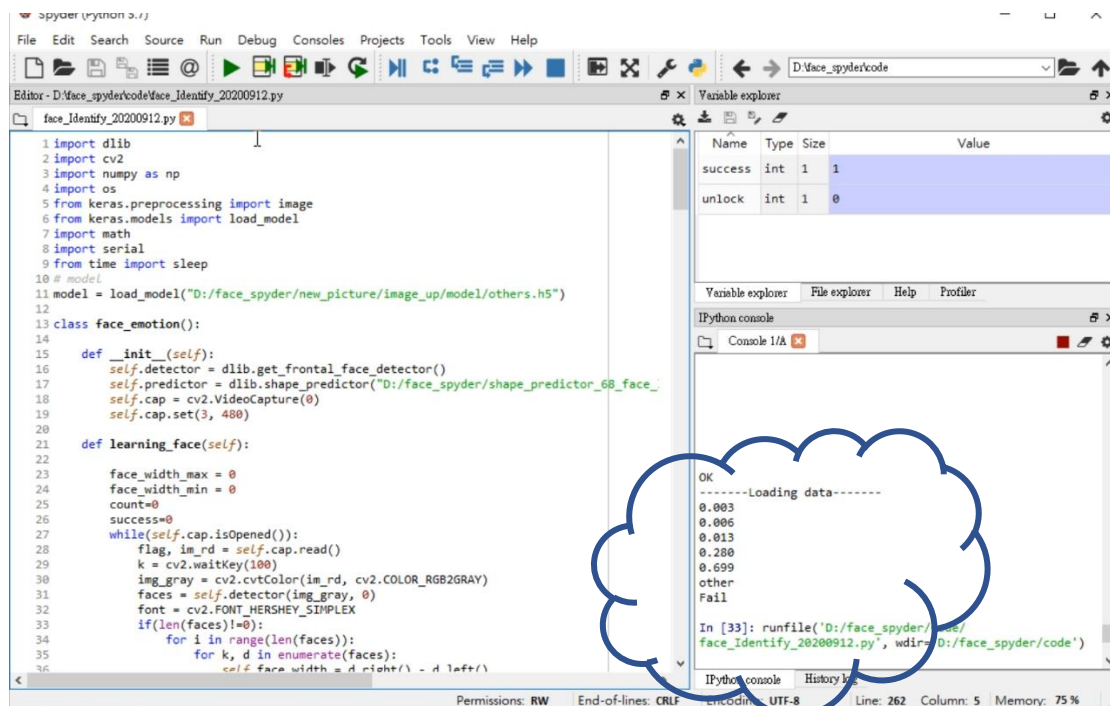
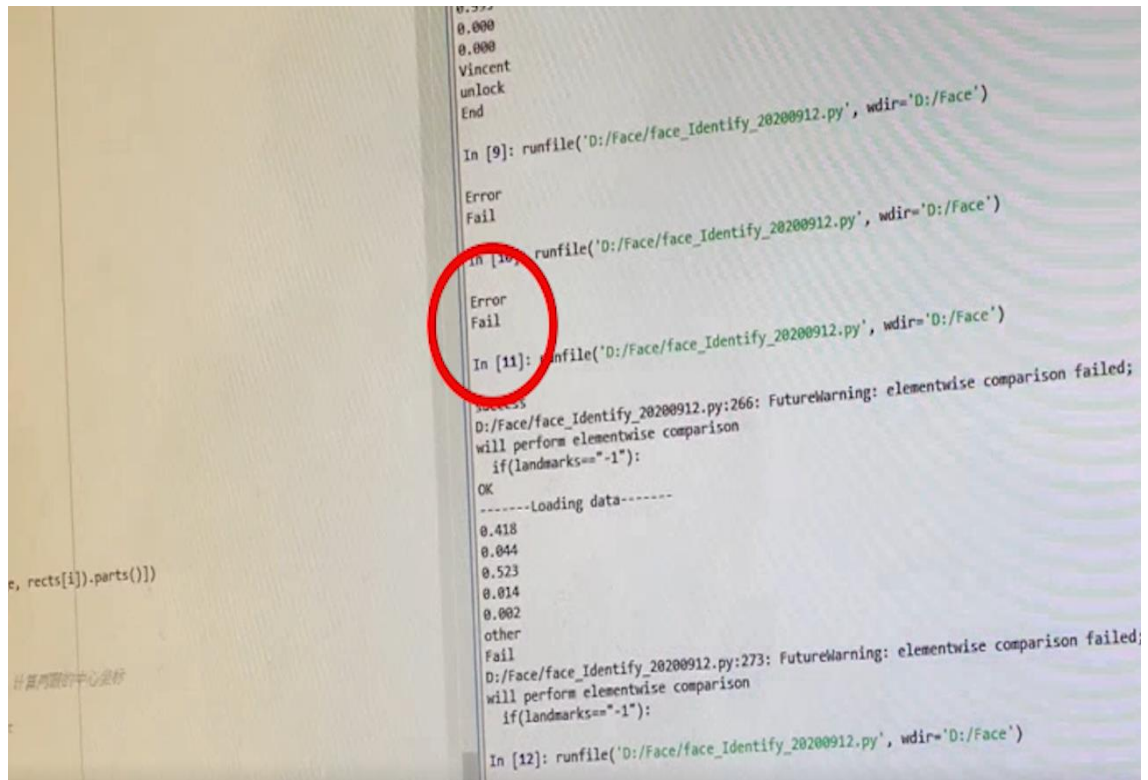


圖 4.24 電腦畫面 (失敗)

如果連活體辨識都沒有過關，直接顯示 Error 然後解鎖失敗。



```
0.000
0.000
Vincent
unlock
End
In [9]: runfile('D:/Face/face_Identify_20200912.py', wdir='D:/Face')
Error
Fail
In [10]: runfile('D:/Face/face_Identify_20200912.py', wdir='D:/Face')
Error
Fail
In [11]: runfile('D:/Face/face_Identify_20200912.py', wdir='D:/Face')
Success
D:/Face/face_Identify_20200912.py:266: FutureWarning: elementwise comparison failed;
will perform elementwise comparison
if(landmarks=="-1"):
OK
-----Loading data-----
0.418
0.044
0.523
0.014
0.002
other
Fail
D:/Face/face_Identify_20200912.py:273: FutureWarning: elementwise comparison failed;
will perform elementwise comparison
if(landmarks=="-1"):
In [12]: runfile('D:/Face/face_Identify_20200912.py', wdir='D:/Face')
```

圖 4.25 電腦畫面（活體辨識失敗）

下圖 4.26 程式碼是將活體辨識過關後，成功拍下的照片與訓練出來的模型比對後，預測其機率，並顯示為誰其機率最高。

```
def face_prediction(success):
    if(success==1):
        path= "C:/Users/as100/Downloads/20200528/prediction/"
        face_cut(path,path)
        images = load_data(path)
        images /=255
        predictions = model.predict_on_batch(images)
        #predictions = model.predict_classes(images)

        print('%.3f' %(predictions[0,0]))
        print('%.3f' %(predictions[0,1]))
        print('%.3f' %(predictions[0,2]))
        print('%.3f' %(predictions[0,3]))
        print('%.3f' %(predictions[0,4]))
        if( predictions[0,0] >= 0.7):
            print('Borong')
        elif( predictions[0,1] >= 0.7):
            print('Hao')
        elif( predictions[0,2] >= 0.7):
            print('Vincent')
        elif( predictions[0,3] >= 0.7):
            print('Teng')
        else:
            print('other')
```

圖 4.26 預測機率部分程式碼

下圖 4.27 程式碼為人臉辨識執行與連接 Arduino 端程式碼，讓已接上 Arduino 的陽極鎖在通過正確人臉是解開門鎖，反之則繼續上鎖。

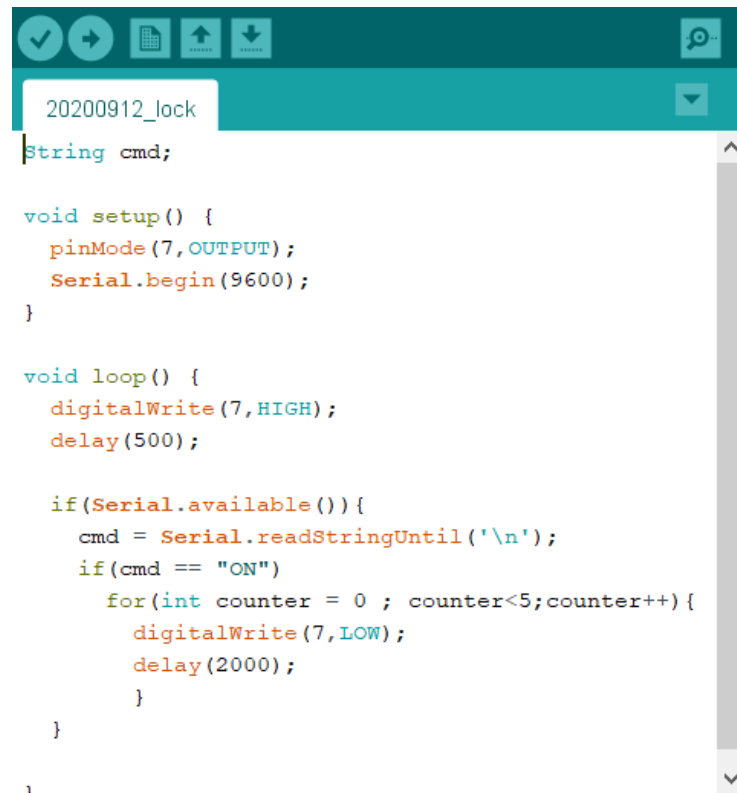
```
if __name__ == "__main__":
    my_face = face_emotion()
    success = my_face.learning_face()
    if(success==1):
        print("success")
    else:
        print("Error")
    unlock = face_prediction(success)
    COM_PORT = 'COM3'
    BAUD_RATES = 9600
    com = serial.Serial(COM_PORT,BAUD_RATES)
    if(unlock==1):

        print("unlock")
        com.write(b'ON\n')
        sleep(2)
        com.write(b'ON\n')
        sleep(2)
        print('End')
    else:
        print("Fail")

    com.close()
```

圖 4.27 人臉辨識執行與解開陽極鎖部分程式碼

下圖 4.28 程式碼是讓其陽極鎖利用 Arduino 配合繼電器能順利給予電力讓 Python 端能控制。



```
String cmd;

void setup() {
  pinMode(7, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(7, HIGH);
  delay(500);

  if(Serial.available()){
    cmd = Serial.readStringUntil('\n');
    if(cmd == "ON")
      for(int counter = 0 ; counter<5;counter++){
        digitalWrite(7, LOW);
        delay(2000);
      }
  }
}
```

圖 4.28 門鎖相關 Arduino 程式碼

4.2.3 熱像儀

熱像儀的部分已利用樹莓派結合熱像儀攝影機，並且配合程式顯示溫度畫面，此裝置會擺設在此次人臉辨識門禁系統旁一同實現。

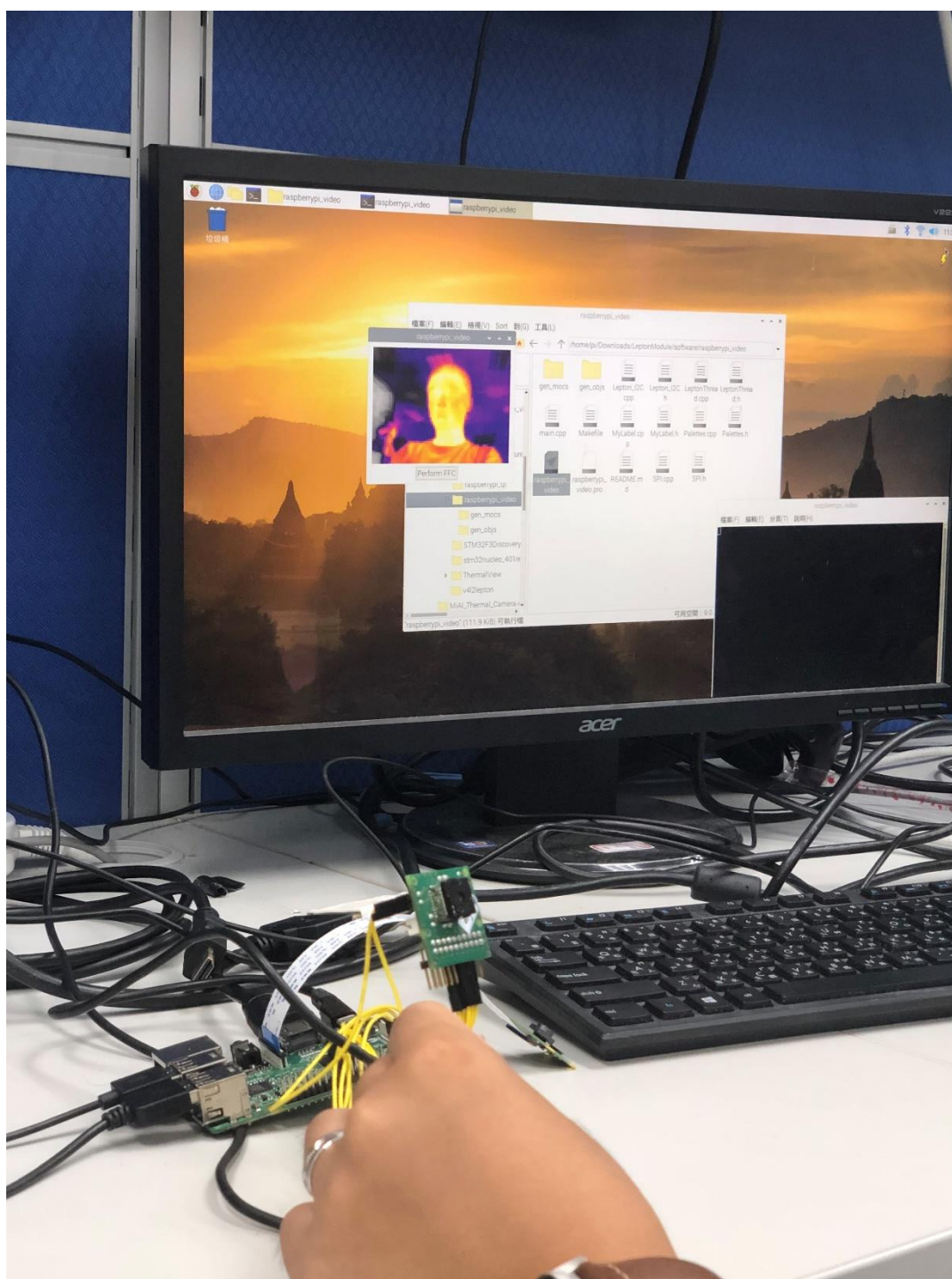


圖 4.29 熱像儀完整畫面

第 5 章 人力配置

參與本次專題人臉辨識門禁系統相關之組員和各分配工作內容如表 5.1 所見。

表 5.1 參與本次專題組員與工作內容分配

姓名	工作內容
王騰緯	資料庫存取、影像處理、訓練模型統整、數據分析
陳柏榮	影像正規化與浮點化、活體辨識發想、Arduino 程式撰寫
邱奕融	影像處理、訓練模型、資料整理、報告撰寫、整理想式
許家豪	辨識與預測、訓練模型、思考方法、硬體管理

第 6 章 結論

本次專題人臉辨識門禁系統經過許多研究之後，得到了不少的改進，因此有幾項目標有不錯的成果。

關於本次專題，實作過程後獲得以下幾點：

(1) 照片品質更佳

原本以筆記型電腦上的攝像頭拍攝下來的照片作為人臉資料，但在畫質上並沒有特別好，後來取得了 HD 攝像頭之後，改進了問題，並以此機器拍下照片為訓練資料。

(2) 經過影像處理的照片

一開始以含有背景的人臉照片進行訓練，但辨識率很差，特徵完全不足，幾乎都辨識錯誤且不是自己，並且有著大家處在相同背景時可能會訓練到背景的問題，經過找尋資料並思考過後，把所有人的照片先擺正後再進行去背而剩下人臉部分，解決了前述很大的問題。

(3) 深度學習與人臉辨識的加強

深度學習以 CNN 作為模型，經過前述的影像處理和細節調整後，除了辨識出來的圖片結果辨識率高之外，幾乎也都對應到對的人，其後還有陌生人問題，與指導老師討論後的結果是建立亞洲人臉資料庫 2000 張當作其他人，並同時和我們四個人一起訓練，並希望能持續改進問題，而人臉辨識部分，活體辨識這項方式也達成預期想要解決安全性問題之一。

(4) 硬體整合軟體完整性

一開始原本想要以市面上的門鈴鎖結合我們此次人臉辨識門禁系統，但嘗試整合後發現，硬體的相容性與我們的軟體並不完美，所以改利用 Arduino、麵包板以及陽極鎖來實現，嘗試過後發現既簡單又快速，實現硬體與軟體的完整性。

參考文獻

- [1] Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2014, September). Facial landmark detection by deep multi-task learning. In *European conference on computer vision* (pp. 94-108). Springer, Cham.
- [2] 數字提取特徵: <https://kknews.cc/zh-tw/tech/z3qjvkg.html>
- [3] Dlib 人臉 68 點: <https://chtseng.wordpress.com/2018/08/18/face-landmark-alignment/>
- [4] LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.
- [5] 卷積神經網路卷積層:
<https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC5-1%E8%AC%9B-%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1%E4%BB%8B%E7%B4%B9-convolutional-neural-network-4f8249d65d4f>
- [6] 卷積神經網路池化層:
<https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC5-1%E8%AC%9B-%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1%E4%BB%8B%E7%B4%B9-convolutional-neural-network-4f8249d65d4f>
- [7] 卷積神經網路全鏈結層:
<https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC5-1%E8%AC%9B-%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1%E4%BB%8B%E7%B4%B9-convolutional-neural-network-4f8249d65d4f>
- [8] 監督式學習:
<https://blogs.nvidia.com.tw/2018/09/supervised-unsupervised-learning/>
- [9] Bao, W., Li, H., Li, N., & Jiang, W. (2009, April). A liveness detection method for face recognition based on optical flow field. In *2009 International Conference on Image Analysis and Signal Processing* (pp. 233-236). IEEE.
- [10] 活體辨識: <https://www.pyimagesearch.com/2019/03/11/liveness-detection-with-opencv/>
- [11] Tronci, R., Muntoni, D., Fadda, G., Pili, M., Sirena, N., Murgia, G., ... & Roli, F. (2011, October). Fusion of multiple clues for photo-attack detection in face

- recognition systems. In *2011 International joint conference on biometrics(IJCB)* (pp. 1-6). *IEEE*.
- [12] 嘴、眼計算圖: <https://www.iotworldtoday.com/2019/12/06/2020-predictions-black-hats-target-facial-recognition-technology/>
- [13] 羅技 C310 HD 視訊攝像頭規格網站:
https://www.eclife.com.tw/SSD/moreinfo_39515.htm
- [14] Arduino 網站:
<https://www.arduino.cc/>
- [15] Raspberry Pi 網站:
<https://www.raspberrypi.com.tw/>
- [16] FLiR Radiometric Lepton Kit V2 網站:
<https://www.taiwaniot.com.tw/product/flir-dev-kit-%E7%B4%85%E5%A4%96%E7%B7%9A%E7%86%B1%E5%BD%B1%E5%83%8F%E9%96%8B%E7%99%BC%E5%A5%97%E4%BB%B6-%E5%8E%9F%E8%A3%9D%E7%BE%8E%E5%9C%8B%E9%80%B2%E5%8F%A3-%E7%86%B1%E6%88%90%E5%83%8F%E5%84%80/>
- [17] Python: <https://www.python.org/>
- [18] OpenCV: <https://opencv.org/>
- [19] Tensorflow: <https://www.tensorflow.org/>