

海青工商資訊科

專題製作

人臉辨識

指導老師: 張洺瑞

班級: 資訊三善

組長:林鈺翔 9305049

組員:洪彩茗 9305050

謝佳祐 9305067

林思萱 9305070

111年9月12日

摘要

此專題以人臉辨識為主軸，並結合 Line Notify，以 OpenCV 擷取 webcam 的畫面並將影像處理，在與資料庫進行比對，實行人臉辨識，最後在將辨識結果為非的影像傳入到使用者的 Line 中，以此來達到監視防盜的效果，我們主要使用 Python 語言，人臉辨識方面的函式庫使用 OpenCV 與 Dlib。

目錄

摘要.....	2
---------	---

第 1 章 前言

1-1 研究背景.....	5
1-2 研究動機.....	5
1-3 研究目的.....	5

第 2 章 文獻探討

第 3 章 研究方法

3-1 架構圖.....	6
3-2 流程圖.....	7
3-3 使用環境及軟體簡介.....	7
3-5 進度表.....	10
3-6 工作分配.....	11

第 4 章 製作過程

4-1 程式設計.....	11
---------------	----

第 5 章 測試狀況

第 6 章 結論

附錄-資料來源

第一章 前言

1-1 研究背景

生物辨識是利用生物特徵本身的差別來區分各個生物體的一種技術，生物辨識包含了常見的人臉辨識，相關應用如 Face ID、人臉門禁等，指紋辨識，相關應用如 Touch ID、指紋門禁等。在現今疫情的時代下人臉辨識相對於指紋辨識的優勢就增加了許多，包含了不用再進行不必要的接觸，只需將臉對準鏡頭，就可以通關，人的五官都長的不同，即使是雙胞胎也有不一樣的地方，如指紋的不同，聲線、或是生活習慣，就連臉部細節都有不同，透過各種生物辨識技術就可以辨識出雙胞胎之間的區別了。

1-2 研究動機

一名日本網友在推特貼出照片，表示早上醒來發現客廳出現一個陌生人，「雖然當下立刻報警了，但還是感到很害怕」，照片應是從監視器中擷取，只見一名穿著藍色內褲的大叔彎曲著身體，躺在客廳地板上、枕著香蕉抱枕熟睡，還有兩名警察準備把人給叫醒。

網友表示，自己在睡前都會確定門窗是否上鎖，家中也裝設了熱感應器和監視器，完全想不到大叔是如何潛入家裡的，目前大叔也被警察帶走了，經過鑑識人員搜查後也確定沒有任何闖入痕跡，最傻眼的就是睡在客廳籠子裡的狗也完全沒吠叫，還睡得很熟。

原PO最後也提醒所有人「不只要在睡前檢查家中門窗是否確實上鎖，平常沒在使用的房間也要記得查看」，不少網友看到後都覺得實在太恐怖，「會不會他一直都躲在你家裡」、「其實他一直都住在閣樓或天花板裡」、「真的太可怕了」。

圖 1：三立新聞網(2019/03/25 15:31)。國際中心綜合報導
<https://www.setn.com/m/ampnews.aspx?NewsID=517200>

我們發現網路上有許許多多的陌生人躲在家裡，卻好一段時間沒被發現的新聞，要防範陌生人就需要有網路監視器之類的設備，然後當有陌生人時就提出警告，但如果連認識的人都警告，很快就會讓人不想管，甚至直接關掉，如果能讓他只有在出現陌生人時才提出警告，那就不會讓人覺得疲乏，也能達到預防陌生人的效果。

1-3 研究目的

利用人臉辨識與 Line Notify 的結合，在發現陌生人時發送訊息來提醒用戶家裡遭陌生人進入，以防範小偷入內行竊的行為，為了達到即時人臉辨識技術，利用少量且重要的特徵，如人臉五官特徵(眼睛和嘴巴)達到理想的辨識率。分析比較不同取樣特徵的辨識結果，找出理想的人臉特徵，最後運用人臉辨識系統來辨識家門口的人臉，方便整理資料。

第二章 文獻探討

八拓科技行銷 - SEO 公司 (2022 年 9 月 22 日) 。 SPYDER 科學的
PYTHON 開發環境- 八拓 。

<https://www.keywordseo.com.tw/blog/spyder-scientific-python-development-environment/>

HackMD (2022 年 9 月 22 日) ， OpenCv 教學目錄 - HackMD

<https://hackmd.io/@LinRoger/BJUMHpyBv>

程式人生 (2022 年 9 月 22 日) 。 Dlib 介紹 。

<https://www.796t.com/content/1547287507.html>

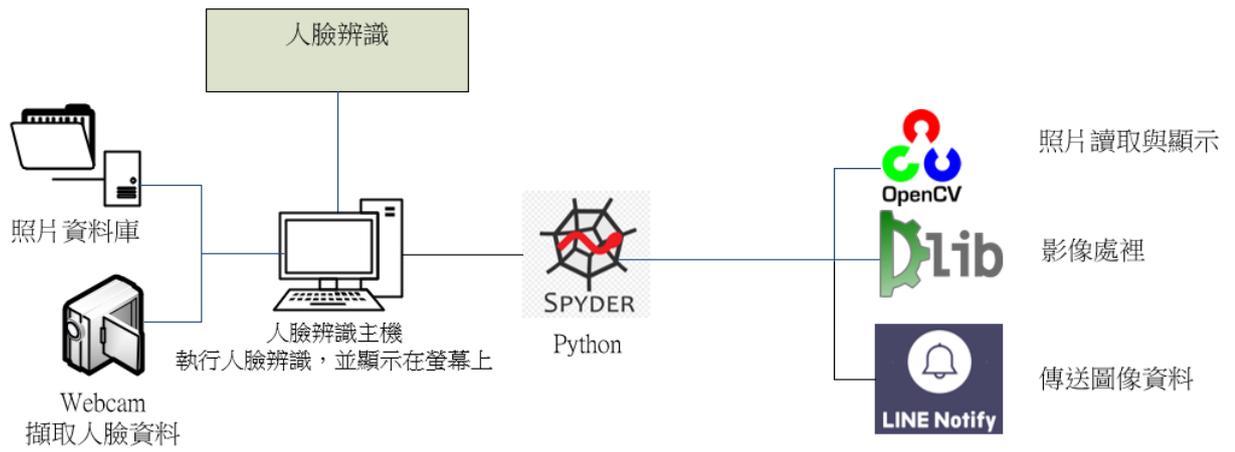
Line Engineering (2022 年 10 月 7 日) 。藉由命令列模式使用 LINE Notify 發送訊息到 LINE 。 <https://engineering.linecorp.com/zh->

hant/blog/using-line-notify-to-send-messages-to-line-from-the-command-line/

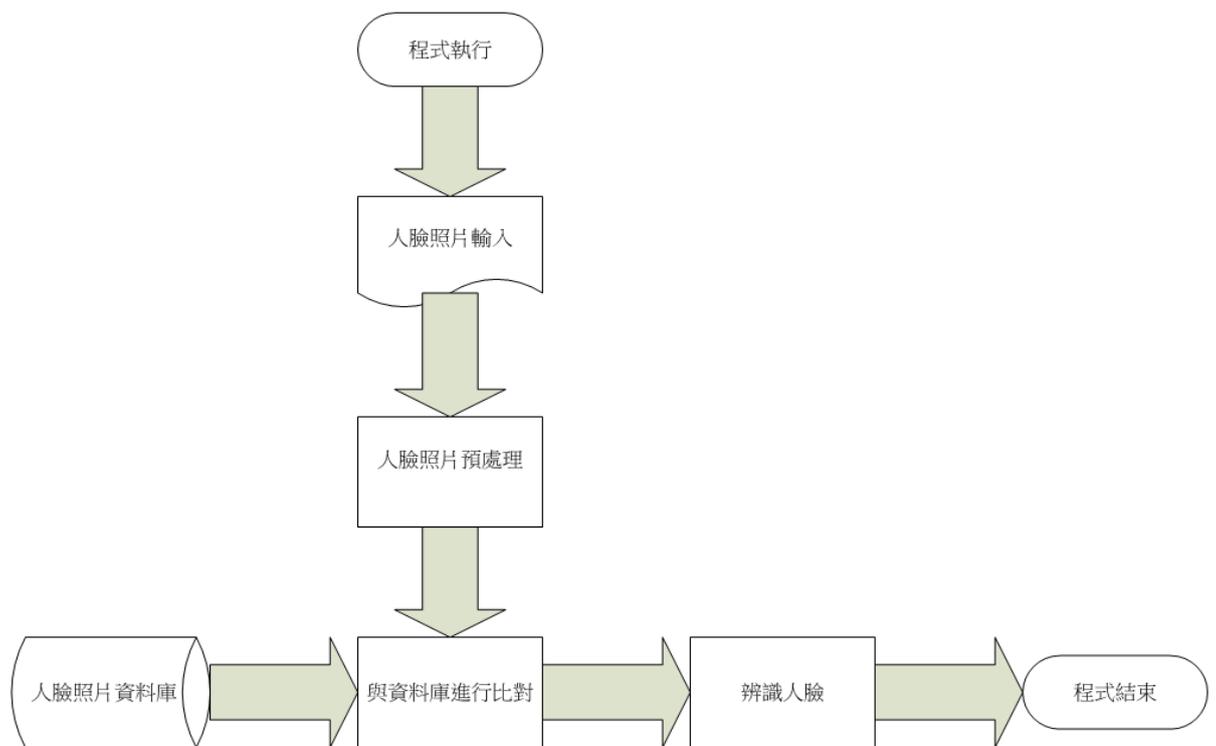
第三章 研究方法

我們預計使用 Python 語言與相關的函示庫來做人臉辨識，函示庫包含了 OpenCV、Dlib、numpy，等影像處理相關，訊息傳送方面我們使用了 Line Notify 來做為訊息的傳送手段，

3-1 架構圖



3-2 流程圖

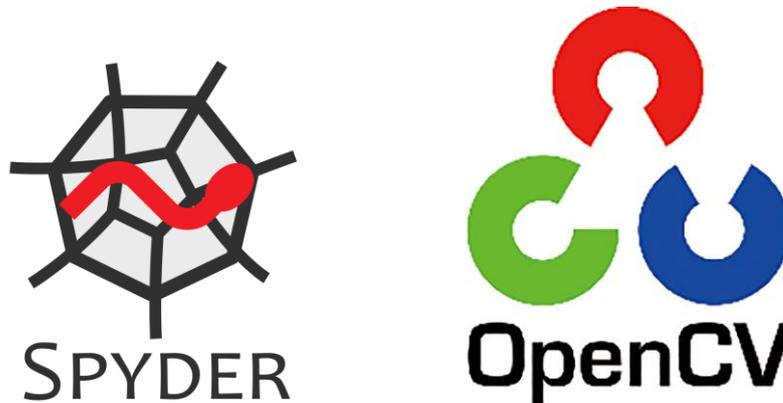


3-3 使用環境及軟體簡介

Spyder (Python 3.6)

Spyder (前身為 Pydee) 是一個使用 Python 語言的開放原始碼跨平台科學運算整合開發環境(IDE)。Spyder 整合了 NumPy, SciPy, Matplotlib 與 IPython, 以及其他開源軟體。

與其他科學數值分析專用 IDE (如 Matlab 或 RStudio) 相比, Spyder 有下列特色: 開放原始碼, 以 Python 編寫並且可以相容於非自由軟體授權。Spyder 可以使用附加元件擴充, 內建互動式工具以處理數據。跨平台的特性使得它可以通過 Anaconda, Winpython 和 Python (x,y) (Windows 平台)。此外在主流的 Linux 發行版本例如 Ubuntu、Debian、Fedora、OpenSUSE 等等中都有它。Spyder 還可以通過繫結 PyQt 或 PySide 來使用 Qt。



OpenCV 4.1.1

OpenCV 的全稱是 Open Source Computer Vision Library, 是一個跨平台的電腦視覺庫。是由英特爾公司發起並參與開發, 以 BSD 授權條款授權發行, 可以在商業和研究領域中免費使用。OpenCV 可用於開發即時的圖像處理、電腦視覺以及圖型識別程式。該程式庫也可以使用英特爾公司的 IPP 進行加速處理。

OpenCV 用 C++ 語言編寫, 它的主要介面也是 C++ 語言, 但是依然保留了大量的 C 語言介面。該庫也有大量的 Python, Java and MATLAB/OCTAVE (版本 2.5) 的介面。這些語言的 API 介面函式可以透過線上文件取得。現在也提供對於 C#, Ch, Ruby 的支援。所有新的開發和演算法都是用 C++ 介面。一個使用 CUDA 的 GPU 介面也於 2010 年 9 月開始實現。

OpenCV 能夠處理的包括: 擴增實境、臉部辨識、手勢辨識、人機互動、動作辨識、運動跟蹤、物體辨識、圖像分割、機器人

Dlib 19.8.1

Dlib 是一個現代化的 C++ 工具箱，其中包含用於在 C++ 中建立複雜軟體以解決實際問題的機器學習演算法和工具。它廣泛應用於工業界和學術界，包括機器人，嵌入式裝置，行動電話和大型高效能運算環境。Dlib 的開源許可證 允許您在任何應用程式中免費使用它。

Dlib 有很長的時間，包含很多模組，近幾年作者主要關注在機器學習、深度學習、影象處理等模組的開發。



Line notify

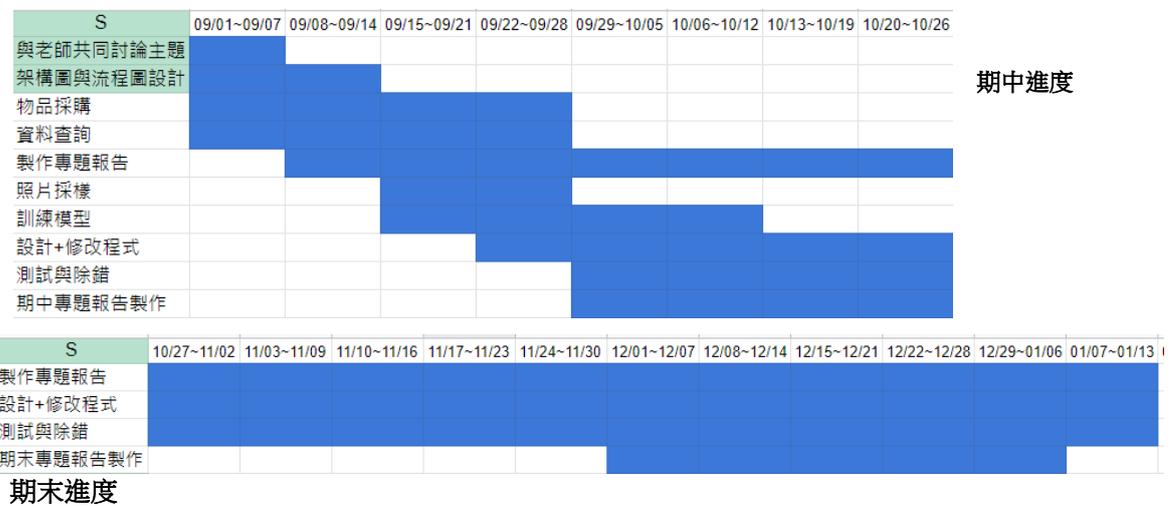
LINE Notify 是 LINE 在 2016 年推出的一個服務，是一種有特殊功能的 API，透過 LINE Notify 產生 personal tokens，並以此發送 HTTP POST 到 API 端點，讓

LINE Notify 發送一則訊息到用戶指定的群組，或者是 1 對 1 接收 LINE Notify 的



訊息

3-4 進度及進度表



3-5 工作分配

程式製作	林鈺翔、謝佳祐
專題製作	洪彩茗、謝佳祐、林鈺翔
簡報製作	林思萱、林鈺翔、洪彩茗、謝佳祐

第四章

4-1 人臉辨識基本架構

1. 提取人臉樣本

程式碼

```
import cv2

camera = cv2.VideoCapture(0)

face_classifier = cv2.CascadeClassifier(
    cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

face_id = input('\n User data input, Look at the camera and wait ...')

count = int(input("Please enter start index: "))
tot = count + 1000

while True:
    success, img = camera.read()
    if success is True:
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    else:
        break

    faces = face_classifier.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0))
        count += 1
        img_name = "data/User."+str(face_id)+'_'+str(count)+'.jpg'
        print(img_name)
        cv2.imwrite(img_name, gray[y:y+h,x:x+w])
        cv2.imshow('image',img)

    if cv2.waitKey(1) == ord('q'):
        break
    elif count == tot:
        break

camera.release()
cv2.destroyAllWindows()
```

執行結果



2. 建立人臉模型

程式碼

```

import os
import cv2
import numpy as np
from PIL import Image
path = 'data'
recog = cv2.face.LBPHFaceRecognizer.create()
name_map = {'Unknown':0, 'chris':1}
detector = cv2.CascadeClassifier(
    cv2.data.harcascades +
    'haarcascade_frontalface_default.xml')
def get_images_and_labels(path):
    image_paths = [os.path.join(path,f) for f in os.listdir(path)]
    face_samples = []
    ids = []
    for image_path in image_paths:
        img = Image.open(image_path).convert('L')
        img_np = np.array(img,'uint8')
        if os.path.splitext(image_path)[-1].split(".")[1] != 'jpg':
            continue
        id = 3
        name = os.path.splitext(image_path)[-1].split(".")[3]
        if name in name_map:
            id = name_map[name]
        faces = detector.detectMultiScale(img_np,1.3,5)
        for(x,y,w,h) in faces:
            face_samples.append(img_np[y:y+h,x:x+w])
            ids.append(id)
    return face_samples,ids
print('Training...')
faces,ids = get_images_and_labels(path)
recog.train(faces,np.array(ids))
recog.save('trainer/trainer.yml')
print("Success!You have finished training. Go on and Continue!")

```

3.開始人臉辨識

程式碼

```

import cv2
import pyautogui
import requests
import time
import datetime

url = 'https://notify-api.line.me/api/notify'
token = 'UKrpnLB0n04DFkNhsMiQ7k9xKdpIJVwmSHj4w1lnNjo'
headers = {
    'Authorization': 'Bearer ' + token
}
data = {
    'message': '是陌生人'
}

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')

face_cascade = cv2.CascadeClassifier(
    cv2.data.harcascades +
    'haarcascade_frontalface_default.xml')

font = cv2.FONT_HERSHEY_SIMPLEX

idnum = 0
isun = 0

names = ['Unknown', 'chris']

cam = cv2.VideoCapture(0)
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(
        gray,
        scaleFactor = 1.3,
        minNeighbors = 5,
        )

    for(x,y,w,h) in faces:
        nowtime = int(datetime.datetime.now().strftime("%F"))
        if nowtime%2==0:
            cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
            idnum, confidence = recognizer.predict(gray[y:y+h, x:x+w])
            nowtime = int(datetime.datetime.now().strftime("%S"))

            if confidence < 100:
                idum = names[idnum]
                confidence = format(round(confidence))
                isun = 0
            else:
                idum = "unknown"
                confidence = format(round(100-confidence))
                cv2.imwrite(r'C:\Users\佳祐\OneDrive\桌面\copy\linenotify\unknown.jpg', img)
                isun = isun+1

            if isun == 50:
                image = open(r'C:\Users\佳祐\OneDrive\桌面\copy\linenotify\unknown.jpg', 'rb')
                imageFile = {'imageFile': image}
                data = requests.post(url, headers=headers, data=data, files=imageFile)

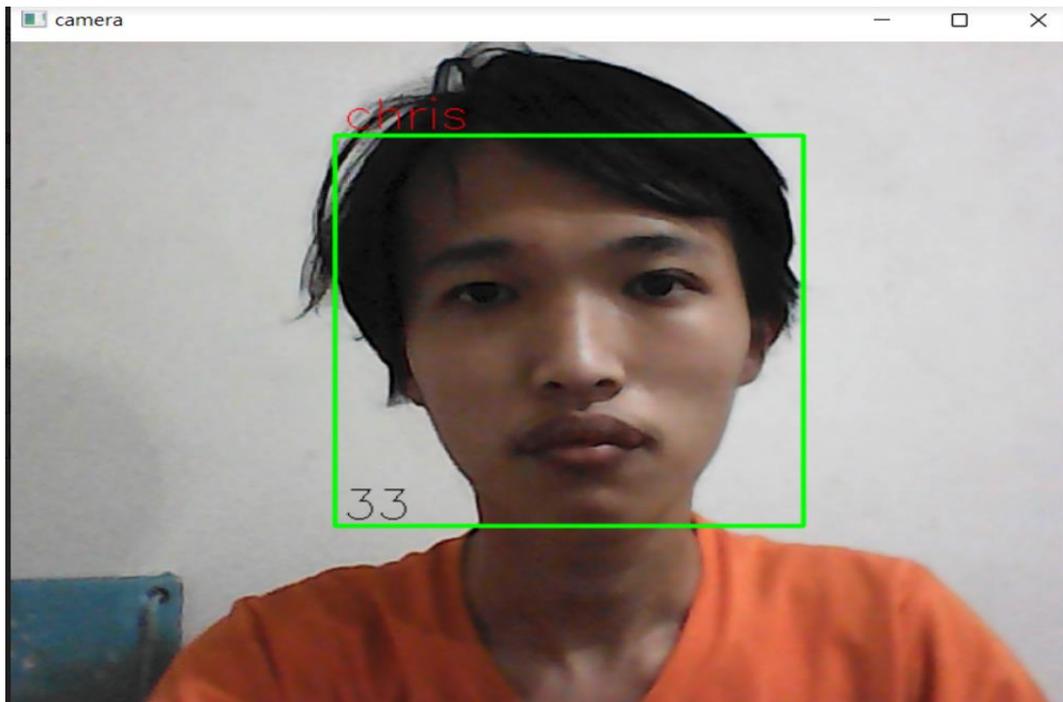
            cv2.putText(img, str(idum), (x+5, y-5), font, 1, (0, 0, 255), 1)
            cv2.putText(img, str(confidence), (x+5, y+h-5), font, 1, (0, 0, 0), 1)

            cv2.imshow('camera', img)
            k = cv2.waitKey(20)
            if k == 27:
                break

        if cv2.waitKey(1) == ord('q'):
            break
    cam.release()
cv2.destroyAllWindows()

```

執行結果



4.向 line 傳送訊息

執行結果



