**崑山科技大學**

**資訊工程系**

**學生專題製作報告**

**CNN模型判別肺炎X光影像Web平台系統**

**Identifying Covid-19 Through Chest X-ray Interpretation in the CNN Computing Platform.**

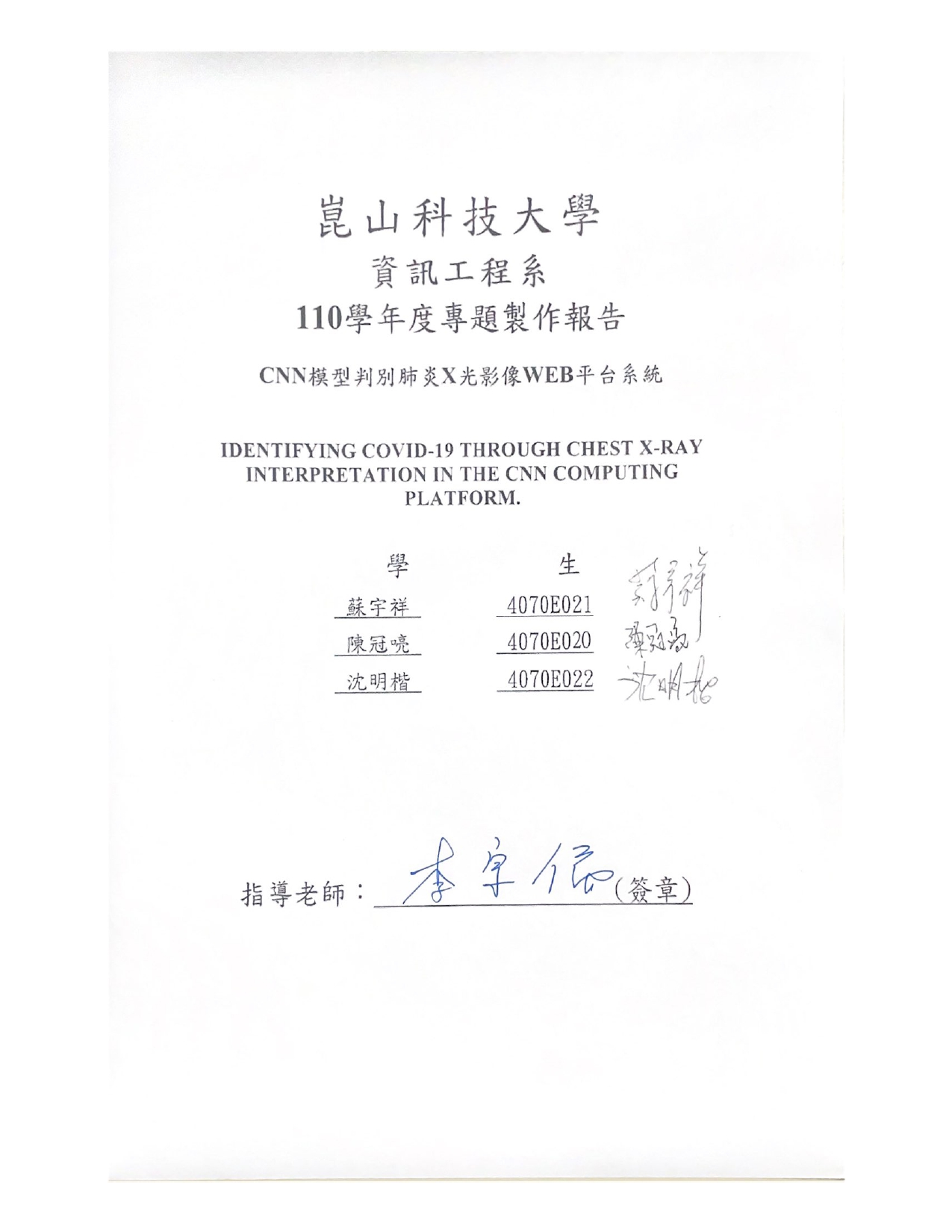
**指導教授：李宗儒**

**專題組員1：蘇宇祥 學號：**4070E021

**專題組員2：陳冠喨 學號：**4070E020

**專題組員3：沈明楷 學號：**4070E022

**中華民國 111年 05 月**

****

**CNN模型判別肺炎X光影像Web平台系統**

蘇宇祥 陳冠喨 沈明楷

崑山科技大學資訊工程系

摘 要

近年興起的人工智慧（Artificial Intelligence, AI）的發展日漸成熟,能透過多種面向來輔助醫療，為了能分攤醫療人員的負擔與辛勞，本組透過AI輔助醫療，利用卷積神經網路（Convolutional Neural Network, CNN）為主要機器學習的方法，來達到決策的行為，其目的是為了輔助醫師診斷，節省人力時間，降低醫護人員負擔，也能夠實現遠距醫療，讓較缺乏醫療資源的偏遠地區能夠有同等資源，不需花費人力與時間看診而耽誤就醫。

本組在此專題上判別肺炎X光症狀為正常(Normal)或肺炎(pneumonia)的準確率有80%，在此專題我們上發現辨識成功的機率並沒有我們符合我們的預期，資料集(Dataset)的不足、訓練次數不足是辨識率需要再提升很重要的原因;在未來我們也希望能增加新冠肺炎(COVID-19)的訓練集，能夠為這場病毒持久戰盡一份心力，早日回到自由呼吸、自由出門的生活。

**關鍵詞：人工智慧、AI輔助醫療**

**Identifying Covid-19 Through Chest X-ray Interpretation in the CNN Computing Platform.**

**SU,YU-XIANG**

**CHEN,GUAN-LIANG**

**SHEN,MING-KAI**

**Department of Information Engineering, Kun Shan University**

Abstract

In recent years, the development of artificial intelligence (AI) has become more and more mature, and it can assist medical treatment through various aspects. Network, CNN) is the main machine learning method to achieve decision-making. The purpose is to assist physicians in diagnosis, save manpower time, reduce the burden on medical staff, and also enable telemedicine, so that remote areas that lack medical resources can With the same resources, there is no need to spend manpower and time to see a doctor and delay medical treatment.

In this topic, our group has an 80% accuracy rate in judging whether the X-ray symptoms of pneumonia are normal or pneumonia. In this topic, we found that the probability of successful identification did not meet our expectations. Dataset and insufficient training times are important reasons why the recognition rate needs to be further improved; in the future, we also hope to increase the training set of Coronavirus disease 2019 (COVID-19), so as to contribute to the war of attrition against the virus and return to the country as soon as possible. To a life of breathing freely and going out freely.

**Key words: Artificial Intelligence, AI-assisted healthcare**

**誌 謝**

在大學專題製作期間，作品與報告書能完成，需要感謝本組指導教授李宗儒老師，在此獻上最大的謝意。專題一開始李宗儒老師就慢慢引導我們，讓我們思考想要做什麼類型的專題，解決哪種狀況等，在製作過程中，也誨人不惓的耐心指導與協助，提供良好的學習環境與資源，更是在我們遇到瓶頸時，給予我們適當的引導。

從當初的摸索到對專題有想法後再實作出來，這段過程當中，雖然我們遇到了許多的障礙，很多時候都讓我們不知自己的方向該如何做，同時也以目前的能力與上課中所學的知識來製作此專題，因應老師所說，專題做完不代表結束，這只是表面的結束，重要的是從中學習到了甚麼，例如適時的報告進度（時間的控管）、與一個上進的態度（愉快的學習）。看看自己所製作的專題，自己也是知道還不夠完美，也沒達到理想的標準，只有不斷精進的能力，才有更好更完美地作品。

**目錄**

[摘 要 i](#_Toc106182352)

[Abstract ii](#_Toc106182353)

[第一章 緒 論 1](#_Toc106182354)

[1.1 研究動機與目的 1](#_Toc106182355)

[1.2 研究方法 1](#_Toc106182356)

[1.2.1 全連接層（Full Connected layer） 1](#_Toc106182357)

[1.2.2 卷積層（Convolution layer） 2](#_Toc106182358)

[1.2.3 池化層（Pooling layer） 2](#_Toc106182359)

[1.2.4 展平層（Flatten layer） 4](#_Toc106182360)

[1.2.5 正規化Softmax函式 4](#_Toc106182361)

[1.3 系統概述 5](#_Toc106182362)

[1.4 工作分配 5](#_Toc106182363)

[第二章 系統相關技術介紹 6](#_Toc106182364)

[2.1 Python 6](#_Toc106182365)

[2.2 Flask 6](#_Toc106182366)

[2.2.1 Flask簡介 6](#_Toc106182367)

[2.2.2 Route路由 6](#_Toc106182368)

[2.2.3 HTTP Methods（GET & POST requests） 6](#_Toc106182369)

[第三章 系統流程與架構 8](#_Toc106182370)

[3.1 系統操作流程 8](#_Toc106182371)

[3.2 系統架構 8](#_Toc106182372)

[3.3 子系統 9](#_Toc106182373)

[3.3 資料庫結構 10](#_Toc106182374)

[第四章 系統功能說明 11](#_Toc106182375)

[4.1 系統語法說明 11](#_Toc106182376)

[4.1.1 url\_for（）路由設定 11](#_Toc106182377)

[4.1.2 flash message 訊息顯示 11](#_Toc106182378)

[4.1.3 extends模板繼承 13](#_Toc106182379)

[4.1.4 Flask-SQLAlchemy資料庫語法 15](#_Toc106182380)

[4.1.5 SQLite管理軟體 16](#_Toc106182381)

[4.2 醫學影像訓練子系統 17](#_Toc106182382)

[4.2.1 醫學影像蒐集 17](#_Toc106182383)

[4.2.2 醫學影像模型訓練 19](#_Toc106182384)

[4.3 使用者Web介面子系統 21](#_Toc106182385)

[4.4 後端Web Server子系統 21](#_Toc106182386)

[第五章 結論 28](#_Toc106182387)

[第六章 參考文獻 29](#_Toc106182388)

**圖目錄**

[圖1. 1卷積層運算過程 2](#_Toc106228276)

[圖1. 2最大池化層示意圖 4](#_Toc106228277)

[圖1. 3平均池化層示意圖 4](#_Toc106228278)

[圖1. 4 展平層示意圖 4](#_Toc106228279)

[圖2. 1 Flask pip安裝套件語法示意圖 6](#_Toc106228287)

[圖3. 1本組系統使用者頁面 8](#_Toc106228513)

[圖3. 2上傳資料進入資料庫 8](#_Toc106228514)

[圖3. 3本專題系統架構圖 9](#_Toc106228515)

[圖3. 4本專題三大子系統架構 9](#_Toc106228516)

[圖3. 5 本專題ERD資料庫關係圖 10](#_Toc106228517)

[圖4. 1 url\_for（）語法示意圖 11](#_Toc106228337)

[圖4. 2 import Flash示意圖 12](#_Toc106228338)

[圖4. 4將迴圈程式新增到欲顯示的html 12](#_Toc106228339)

[圖4. 5前端系統刪除介面的flash message 12](#_Toc106228340)

[圖4. 6繼承語法說明圖 13](#_Toc106228341)

[圖4. 7 本專題home.html父模板 13](#_Toc106228342)

[圖4. 8 本專題upload.html子模板 14](#_Toc106228343)

[圖4. 10 Flask-SQLAlchemy 15](#_Toc106228344)

[圖4. 11 DB Browser軟體示意圖 16](#_Toc106228345)

[圖4. 12由本系統（左）上傳資料到資料庫（右）示意圖 16](#_Toc106228346)

[圖4. 13 Kaggle Open Data頁面[16] 17](#_Toc106228347)

[圖4. 14將Kaggle數據下載到本地電腦 17](#_Toc106228348)

[圖4. 15測試集資料（正常肺部） 18](#_Toc106228349)

[圖4. 16測試集資料（罹患肺炎） 18](#_Toc106228350)

[圖4. 17影像訓練模型增益率（acc為模型訓練精度、val\_acc為驗證精度） 19](#_Toc106228351)

[圖4. 18影像訓練模型遺失率（loss為模型訓練損失值、val\_loss為測試損失） 19](#_Toc106228352)

[圖4. 19模型訓練後產生之模型檔案（提供給伺服網頁使用） 20](#_Toc106228353)

[圖4. 20使用者前端登入介面 21](#_Toc106228354)

[圖4. 21登入後會依帳號顯示醫生名稱於頁面 21](#_Toc106228355)

[圖4. 22本專題辨識肺部X光片介面 22](#_Toc106228356)

[圖4. 23由前端資料上傳至資料庫示意圖 22](#_Toc106228357)

[圖4. 24本專題辨識記錄查詢介面示意圖 22](#_Toc106228358)

[圖4. 25辨識記錄列表示意圖 23](#_Toc106228359)

[圖4. 26 ksu1身分登入頁面示意圖 23](#_Toc106228360)

[圖4. 27編輯該筆資料頁面 24](#_Toc106228361)

[圖4. 30刪除歷史紀錄資料庫示意圖 25](#_Toc106228362)

[圖4. 31系統alert警告是否刪除該筆資料 26](#_Toc106228363)

[圖4. 32 Flash message顯示該筆紀錄已刪除 26](#_Toc106228364)

[圖4. 33該筆資料已被刪除示意圖 27](#_Toc106228365)

**表目錄**

[表1. 1工作分配表 5](#_Toc104163452)

[表4. 1資料庫示意表 15](#_Toc104163460)

第一章 緒 論

1.1 研究動機與目的

新型冠狀肺炎（Covid-19, Corona Virus Disease 2019）近年來肆虐全球，截至2022年5月22日，全球已累計報告逾5.2億例確診個案[1]，其中逾612.1萬人死亡，台灣截止目前確診數累計已破百萬人次。台灣在2020年初就出現第一起病例，然而新型冠狀肺炎症狀有發燒、咳嗽、肌肉痠痛、嗅味覺失調等症狀，更是會讓肺部產生纖維化。新冠肺炎影響急遽，醫療院所為檢疫與掌握病患是否罹患肺炎，傳統的方式是由醫師人工判讀肺部X光影像，有時會造成誤判，很可能導致患者失去黃金醫療時間，現在當科技日新月異，漸漸人工智慧 （Artificial Intelligence, AI） 影像辨識技術結合醫學影像，達到AI輔助醫療，希望可以藉由人工智慧協助，減少醫師工作量，患者能夠及早治療。本研究首先透過醫師拍攝患者胸腔X光，再將患者肺部X光影像檔，透過預先訓練完成之影像辨識模型來進行預測，預測之結果可讓特定醫師根據病患狀況進行相關醫療動作。雖然已有相關的研究與成果，但本作品將以優化的使用者Web介面為特色，並搭配相關的資料儲存和分析技術，提供更完整與方便的操作環境。

1.2 研究方法

本研究以卷積神經網路（Convolutional Neural Network, CNN）為主要機器學習方法。機器學習當中，卷積神經網路是一種前饋神經網路的概念，它裡面的人工神經元可以影響範圍內的單元，最常應用於人聲語音辨識、影像處理以及影像辨識三種領域。卷積神經網路[2]已成為眾多領域的研究方法之一，它是目前深度神經網路 deep neural network 領域的發展主力，由具有可學習權重和偏差的神經元組成，可以直接上傳或是輸入所需的圖像檔，免除了圖像的複雜處理與分類，因此得到研究人員的廣泛使用。而提到CNN卷積神經網路，讓人聯想到深度學習模型主要有全連接（Fully Connected，FC）網絡結構、卷積神經網絡（Convolutional Neural Network，CNN）和循環神經網絡（Recurrent Neural Network，RNN）。它們各有自身的特點，在不同的實驗擁有重要地位。

1.2.1 全連接層（Full Connected layer）

在CNN結構中，經多個卷積層和池化層後，連接著1個或1個以上的全連接

層[3]，全連接層中的每個神經元與其前一層的所有神經元進行全連接。可以整合卷積層或池化層中，具有類別區分性的局部信息，全連接網路就是一個分類器，我們經過數個卷積、池化後的結果進行分類。 [4]

1.2.2 卷積層（Convolution layer）

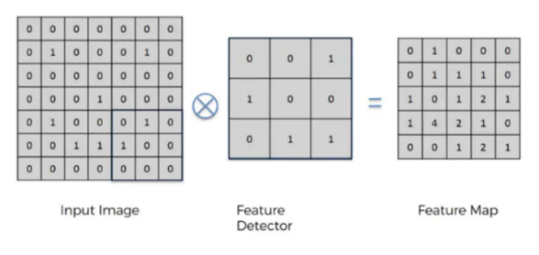
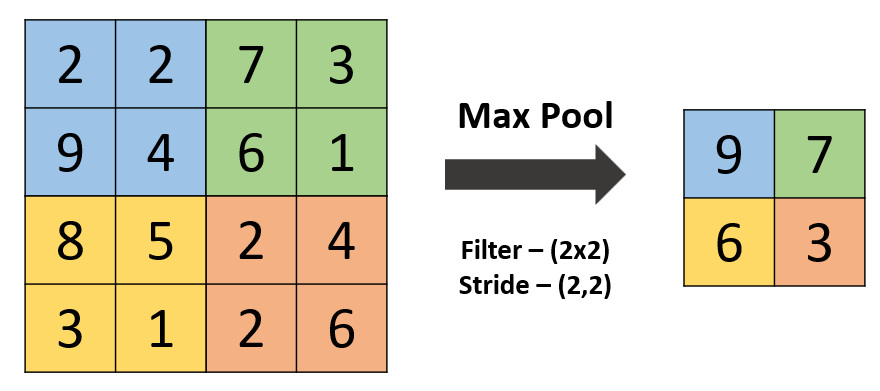
卷積是透過兩個函數 f（） 和 g（） 生成的第三個函數之一種式子，表徵函數 f（） 與經過翻轉和平移的 g（） 的乘積函數所圍成的曲邊梯形面積，意即是先將一個函式翻轉，然後進行滑動疊加。卷積的卷指函示翻轉，代表著g（t）變成g（-t）的過程；而卷積的積指積分或加權求和。而卷積層是由多個特徵面組成的，每個特徵上又有多個神經元組成，它每一個神經元都有相對應的卷積層與上層特徵面的區部連結，卷積層是一個矩陣（對於二維而言可為3\*3、5\*5的矩陣），卷積運算就是將原始圖片的與特定的Feature Detector（filter）做卷積運算（符號⊗）（如圖1.1）。最常用於訊號分析與影像處理。

圖1. 1卷積層運算過程

1.2.3 池化層（Pooling layer）

池化層主要目的就是縮小影像，常用的方式是將長寬各縮小一半處理，主要用於做圖片特徵萃取，擷取圖像當中最大或平均的部分，以減少影像的參數量和計算量，而池化層在輸入的每個深度切片上都是獨立操作。常見的池化方式有Max pooling（最大池化）與 Average pooling（平均池化）兩種。最大池化（Max pooling）是從過濾器覆蓋的特徵圖區域中選擇最大元素。最大池化層之後的輸出將是一個包含前一個特徵圖最突出特徵的特徵圖（如圖1.2）。平均池化（Average pooling）計算特徵圖塊的平均值。平均池化通常在卷積層之後使用，在眾多淤快數據中，提取每區塊計算出的平均值，有別於最大池化層，能更平滑地提取特徵。（如圖1.3）



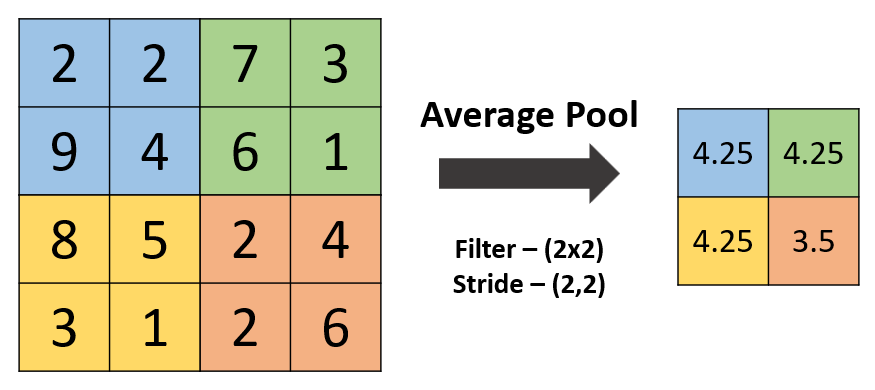
圖1. 2最大池化層示意圖

圖1. 3平均池化層示意圖

1.2.4 展平層（Flatten layer）

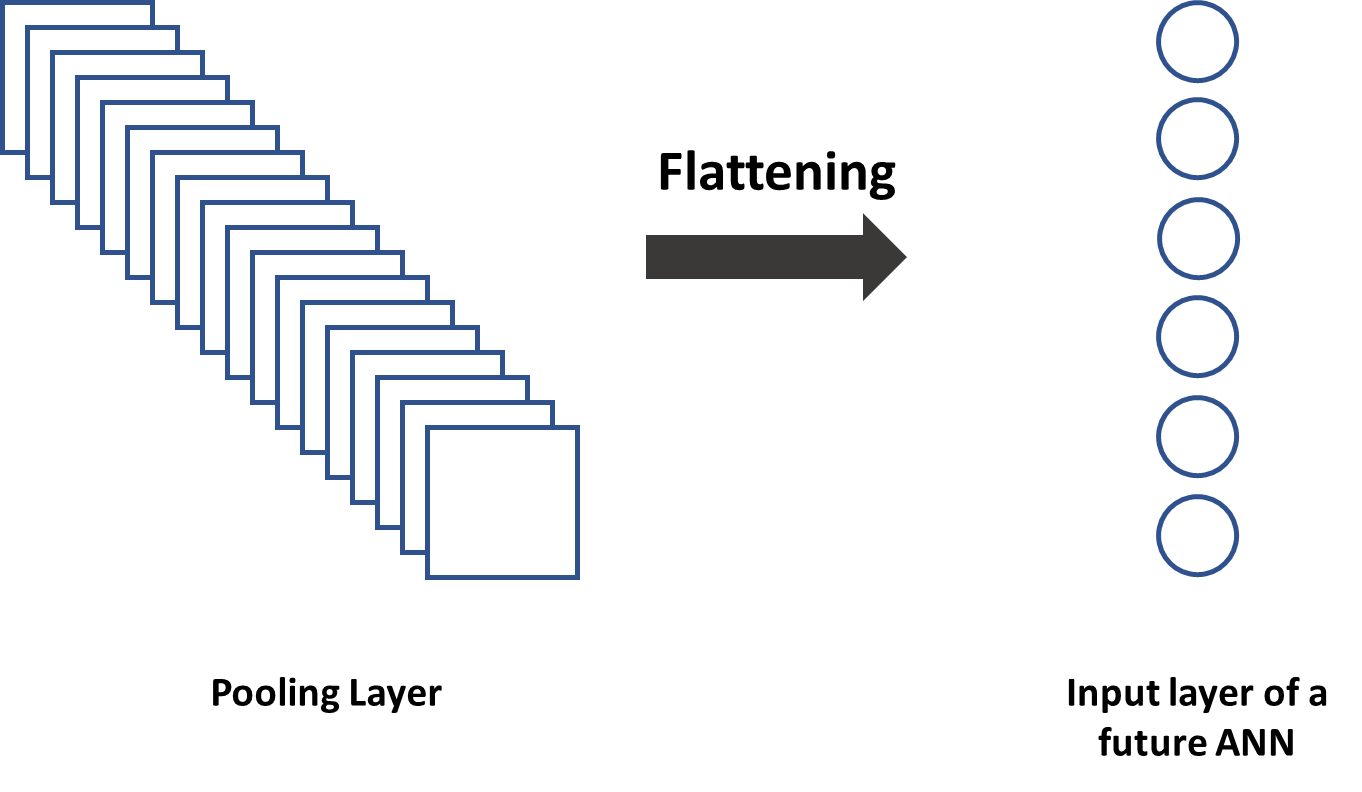
為了能產生全連接層所需的特徵點，可將較小尺寸（如3x3, 5x5）的特徵圖直接展開，把n維的資料統整為一維資料（如圖1.4）。

圖1. 4 展平層示意圖

1.2.5 正規化Softmax函式

影像運算輸出會有多個，但正確者只有一個，為強化該輸出的機率，通常還會搭配一個機率正規化函式。其中Softmax函式主要概念就是把所有輸出的機率加總當成分母，再以原輸出機率當成分子，求出新的機率值，這樣會更接近現實。

1.3 系統概述

本系統擁有三大子系統。分別為醫學影像訓練子系統、使用者Web界面子系統以及網頁伺服Web Server子系統。各組員分別完成三大子系統（如表1.1）

* 醫學影像訓練子系統：到開放式資料網站（Kaggle）蒐集肺部X光影像，包括正常肺部X光與罹患肺炎X光兩種影像，獲取後進行資料處理與影像模型訓練。
* 使用者Web界面子系統：本系統擁有一網頁前端子系統，提供使用者將圖片於任一方上傳，影像透過與置於伺服子系統之模型比對，再將結果回傳於前端告知使用者。使用者有專屬的帳號密碼，提供安全性。這個網站必須要由相關醫師才能進入，而非開放式的網站，讓病患的醫療資訊曝光。
* 網頁伺服Web Server子系統：利用Python Flask撰寫伺服網頁程式，可將模型置於此處，並透過伺服器將前後端整合，將比對結果輸出於前端。

1.4 工作分配

表1. 1工作分配表

|  |  |
| --- | --- |
| 組員 | 內容 |
| 蘇宇祥 | 1. 醫學影像蒐集與CNN模型訓練 2. 前端頁面設計 3. 系統整合 |
| 陳冠喨 | 1. 前端頁面排版設計 2. 醫學影像蒐集 3. 系統整合 |
| 沈明楷 | 1. 後端Web Server程式撰寫 2. SQLite資料庫語法與設定 3. 系統整合 |

第二章 系統相關技術介紹

本專題系統語法是使用Python程式語言，搭配其輕量級網頁框架Flask語法，將整個系統建置與維護。

2.1 Python

Python是現今世界上熱門的一種高階程式語言，其優點有容易上手、許多強大的函式庫及框架，更結合了現在資訊領域最熱門的大數據、機器學習（Machine Learning）及電腦視覺（Computer vision）。

而本組透過Python執行模型訓練、後端使用Python微框架Flask與Flask SQlalchemy使用SQLite來使用資料庫進行銜接。

2.2 Flask

2.2.1 Flask簡介

Flask[8]是Python的輕量Web微框架[9]，是由Jinja2（Jinja也是Ronacher創作，是給Python程式語言的模板引擎，並在BSD許可證下發行。）模板引擎和Werkzeug WSGI（Web Server Gateway Interface）工具箱組成。透過flask可以輕易架設網站，使用CMD（命令提示字元）pip指令來安裝Flask套件。pip 是Python 安裝管理套件工具，所以我們需要用到pip install flask來進行下載（如圖2.1）。

圖2. 1 Flask pip安裝套件語法示意圖

2.2.2 Route路由

使用者可透過自己所定義的路由，來訪問網頁。程式中的@app.route（"/"）是路由的位置（/代表預設首頁），CMD（命令提示字元）輸入flask run來執行，預設網址localhost （http://127.0.0.1:5000/），假設網站是/flask，那@app.route（"/flask"）即可建立網路路由[10]。

2.2.3 HTTP Methods（GET & POST requests）

GET跟POST requests可透過@app.route（）新增method來處理HTTP請求。[11]

* Request:是用來取得數據，需透過from flask import request來import套件。
* GET:將數據傳送，利用『？』作為第一個參數的連接符號，第二個之後的參數是以『&』符號作為連接符號。

POST:由於使用GET方法傳送資料會曝露資訊在網址列，若要使用安全性較高的方式就是使用POST資料表單發送到伺服器。

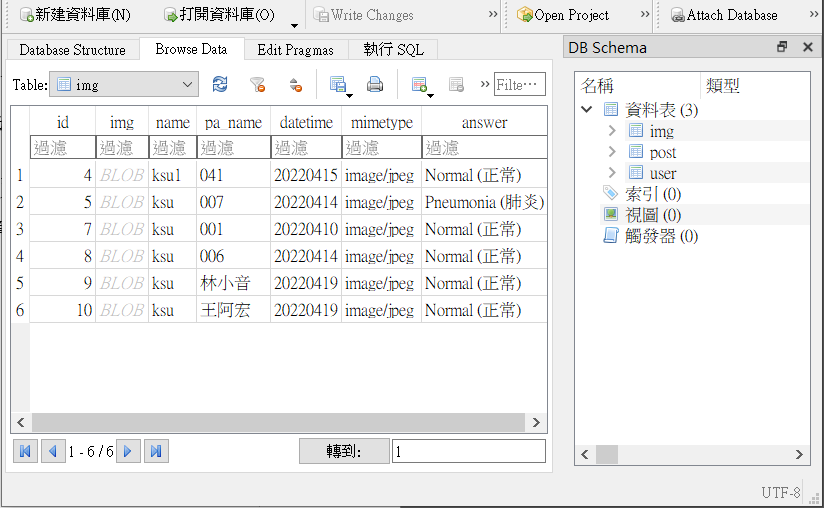
第三章 系統流程與架構

以下說明本專題整體系統流程與架構。本章節提及本專題三大子系統，簡介各個子系統運作狀況與本專題資料庫結構。

3.1 系統操作流程

使用者先登入自己帳號密碼後，點選「選擇圖片」（如圖3.1）後，可由任意處將影像上傳，填入病患相關資料後，即可按下「辨識」，系統即會與後端伺服網頁之模型比對後，將結果顯示於前端，並將資料送進資料庫內（如圖3.2），以備特定醫師查看與後續動作。

圖3. 1本組系統使用者頁面

圖3. 2上傳資料進入資料庫

3.2 系統架構

此系統分為三階段，分別為影像選擇、肺炎辨識與上傳資料庫。影像先由訓練端訓練模型後，提供模型檔案給Web Server放置，再由使用者上傳之影像比對，

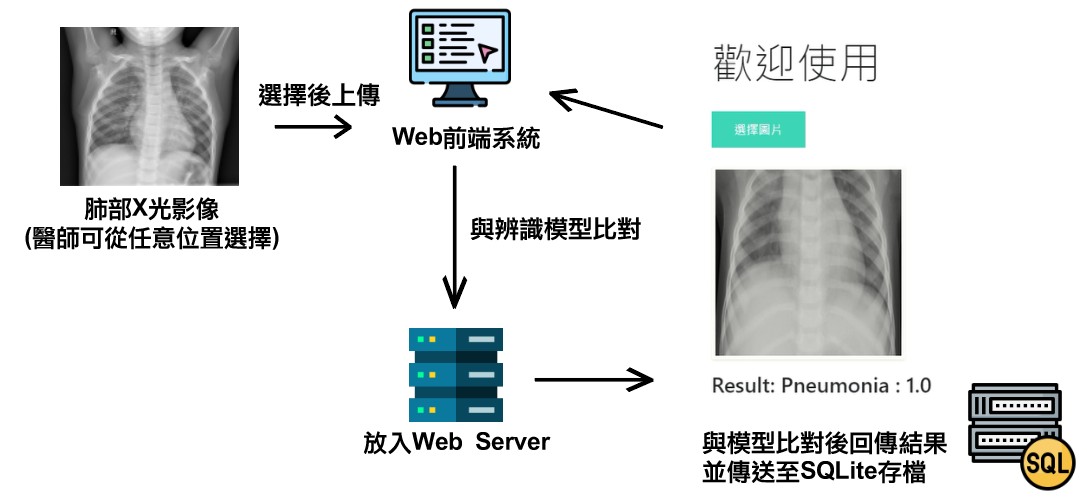
結果顯示與上傳資料庫。（如圖3.3）

圖3. 3本專題系統架構圖

3.3 子系統

本系統共包含3個子系統。子系統『醫學影像辨識訓練模型子系統』肺部影像資料訓練，以建立CNN模型。子系統『使用者Web介面子系統』Web介面選擇影像，透過Web Server連結模型進行判別與比對。子系統『網頁伺服（Web Server）子系統』透過網頁伺服將Web介面連結到CNN模型。（如圖3.4）

圖3. 4本專題三大子系統架構

3.3 資料庫結構

本系統（CNN模型判別肺炎X光影像Web平台系統） ERD（Entity Relationship Diagram）資料庫共有三個table:user、img與post（如圖3.5）

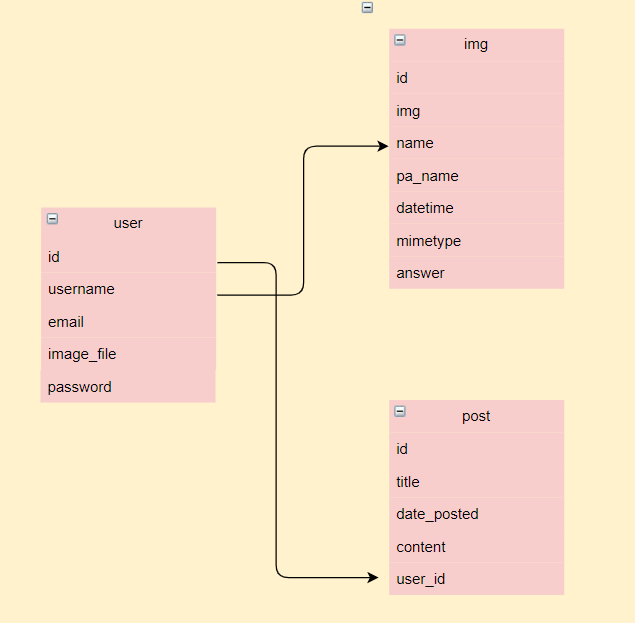
* user負責管理使用者資訊（使用者id、使用者姓名、使用者信箱、使用者頭貼與使用者密碼）
* img負責醫師上傳之X光影像資訊（X光id、x光影像、x光檔案名稱、X光病患姓名、上傳的醫師名稱、上傳日期、檔案名稱與辨識結果）
* post負責醫師上傳聊天室貼文資訊（文章id、文章標題、上傳文章日期、上傳文章內容與上傳使用者的id）

圖3. 5 本專題ERD資料庫關係圖

第四章 系統功能說明

本章節將會提到本專題所使用到的重要語法（而本專題程式基礎建立在Python語言上，前端介面與後端Web Server皆會使用）、使用工具以及各個子系統介紹。

4.1 系統語法說明

Flask可以透過Werkzeug WSGI工具包來建立web框架跟Jinja2網頁模板來進行網頁框架的建立，Flask也有許多功能強大的擴展包，例如:本專題有使用到的Flask-SQLalchemy做資料庫的操作、Flask-WTF表單的渲染與Flask-Login的登入狀態管理等，使Flask微框架的功能更加強大。

4.1.1 url\_for（）路由設定

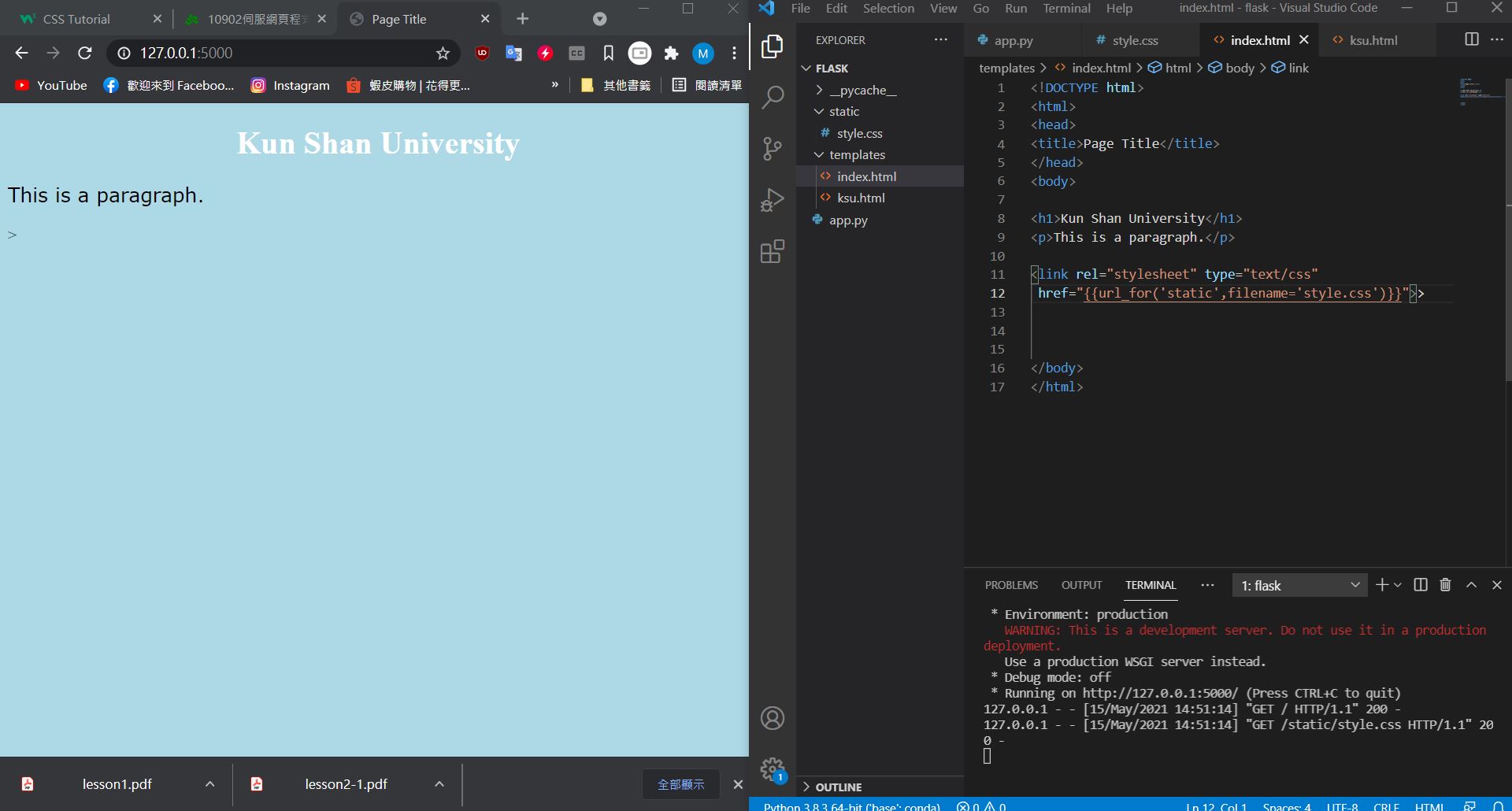
透過url\_for（如圖4.1）取得函數的URL，並非是route路徑。而redirect可以讓flask重新導向頁面，利用url\_for與redirect的搭配我們就可以重新導向指定的頁面。

圖4. 1 url\_for（）語法示意圖

4.1.2 flash message 訊息顯示

當使用者提交資料後，若資料的格式不符合或是要告訴使用者處理流程是否完成，可以透過flash訊息顯示[13]的方式告訴使用者哪個資料填寫錯誤不符合，或者是使用者登入成功，會跳出message來告訴使用者，首先使用者需要先import flash（如圖4.2），flask有用到session需要一組密碼進行secret key加密（如圖4.3），透過迴圈來取得訊息（如圖4.4），來判斷有無message訊息，例如：本專題的刪除功能（如圖4.5），若使用者成功刪除的該筆紀錄，則會顯示綠色的該筆紀

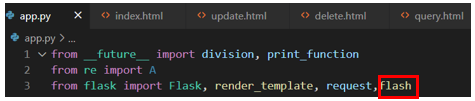
錄已刪除字樣。

圖4. 2 import Flash示意圖

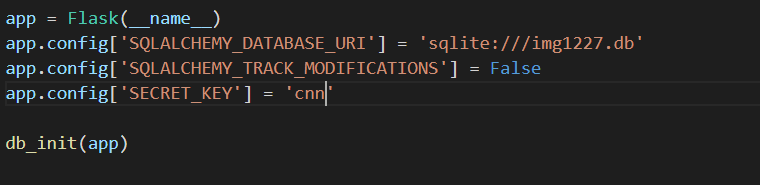
Flask的flash須設定secret\_key加密，密碼可以自己設定自己喜愛的密碼，設定金鑰的原因是因為防止CSRF跨站請求偽造 （Cross-site request forgery），但CSRF又跟常見駭客攻擊手法XSS注入攻擊 （Cross-Site Scripting）不一樣，CSRF是透過網站的cookie與session來登入驗證身分，當我們已經登入驗證過了，之後的操作就不需要再重新登入驗證，讓駭客有機可趁，這就是CSRF可怕的地方，為了防止CSRF的攻擊手段，而有了secret\_key的加密措施。

圖4. 3設定secret key加密示意圖

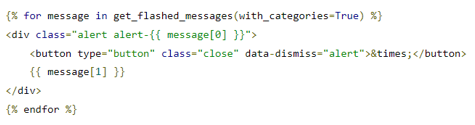
若Flask的flash message要成功顯示在網頁上，則必須要加入下列的迴圈程式，使用get\_flashed\_messages（）函數來取得session所有訊息（message）。

圖4. 3將迴圈程式新增到欲顯示的html

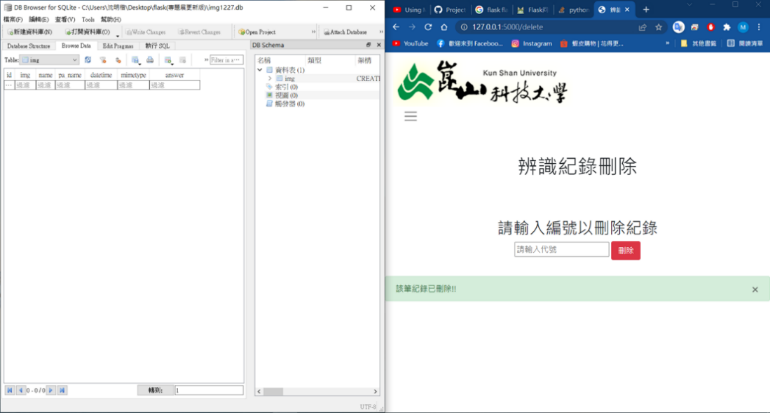
將迴圈程式加到所要顯示的html，例如:下圖顯示我們要刪除紀錄，並使用flash message來提醒使用者是否有刪除掉欲刪除的該筆資料，若成功就會顯示該筆紀錄已刪除；若失敗（無該筆資料在資料庫）會顯示查無此紀錄。

圖4. 4前端系統刪除介面的flash message

4.1.3 extends模板繼承

利用{% block content %}{% endblock %}放父樣板所沒有的內容。使用block區塊是子樣板的關鍵字，content是子模板本體的名稱。在同一個頁面content的名稱不能一樣（如圖4.6）。

圖4. 5繼承語法說明圖

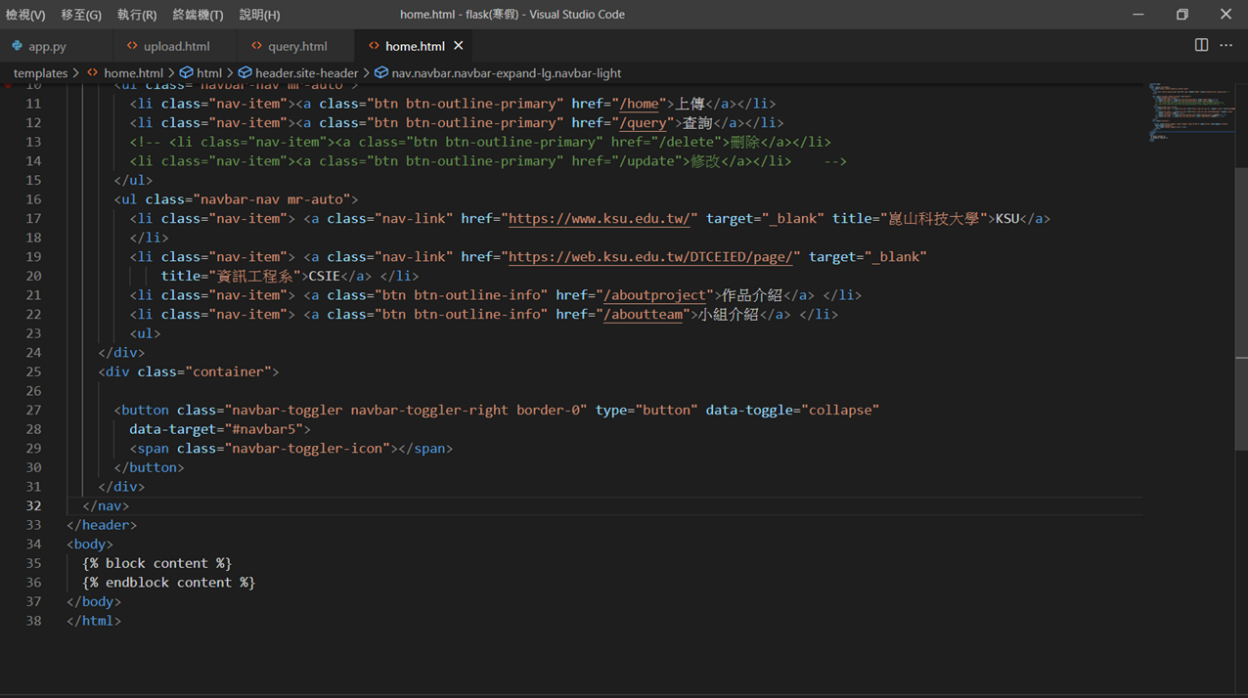
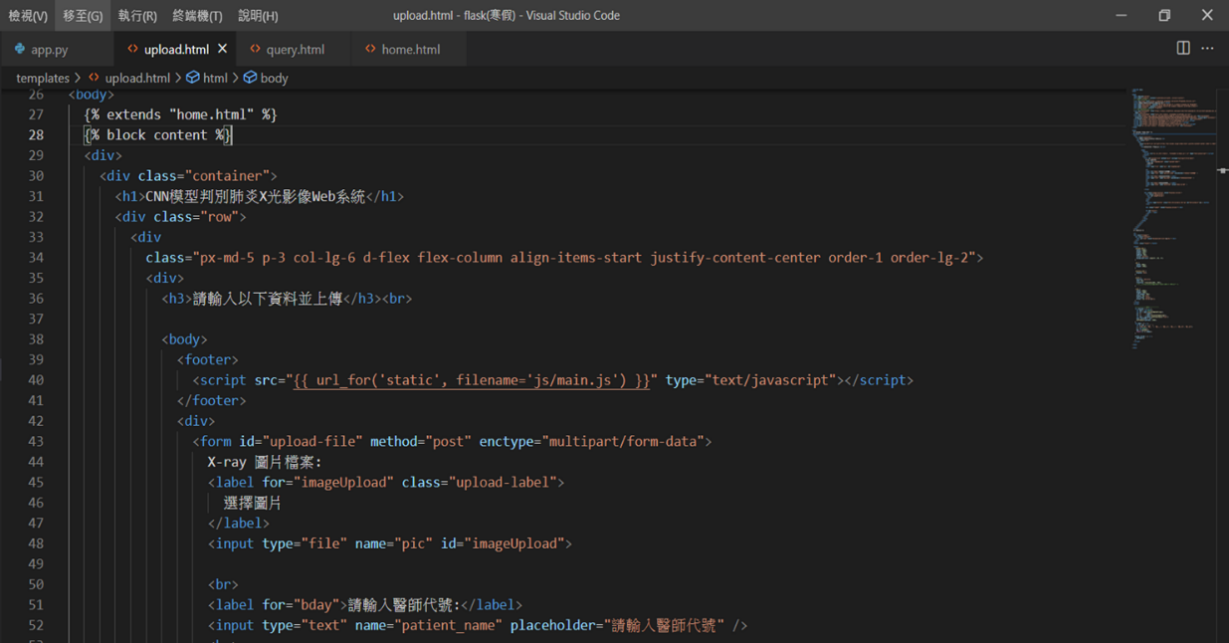
例如：本專題的home.html需要給子模板[14]（upload.html）繼承（extands） 的導覽列navbar，home.html的{% block block content %}{% endblock content %}要接upload.html的{% extends "base.html" %}{% block content%}{% endblock content%}的範圍（如圖4.7、圖4.8、圖4.9）。

圖4. 6 本專題home.html父模板

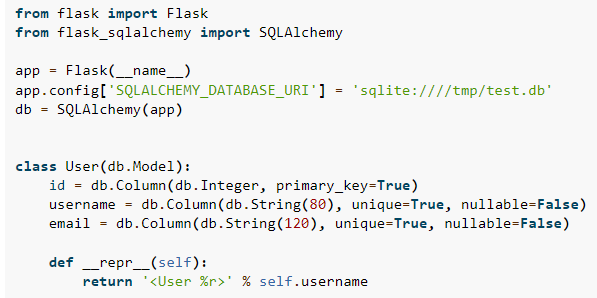
圖4. 7 本專題upload.html子模板

會發現Jinja2的模板繼承（extands）非常強大，透過繼承的方式來取代重複的方式，有點像在子模板預留一個孔洞，將父模板的內容繼承給子模板，就不需要一直複製重複的程式碼，方便又簡潔。以下是本專題導覽列（navbar）繼承的完成圖。

圖4.9 upload.html已繼承home.html的navbar導覽列

4.1.4 Flask-SQLAlchemy資料庫語法

Flask本身不支援SQL操作，SQLAlchemy[13]（如圖4.10）支援物件關係對映器ORM（Object Relational Mapper）框架來使用資料庫，SQLAlchemy可使用SQLite、PostgreSQL、MySQL、Oracle，使用ORM 來操作資料庫，透過pip install flask-sqlalchemy來安裝套件。[15]

圖4. 8 Flask-SQLAlchemy

使用 app.config 來連線到database設定SQLite資料庫的路徑，本專題使用較輕量的資料庫SQLite進行資料庫的操作，下方class去定義資料庫欄位，如上圖所示，這是一個資料表為User的table（表格）id、username、email 為資料庫欄位（Column）。（如表4.1）

* db string的格式（Integer、float、string……等）
* unique = True （設定為唯一性）
* index = True （索引值）
* nullable = True （允許為null空值）
* default = ‘預設值’

表4. 1資料庫示意表

|  |  |  |
| --- | --- | --- |
| id | name | age |
| 1 | Ksu | 18 |
| 2 | Steven | 22 |
| 3 | Grady | 21 |

4.1.5 SQLite管理軟體

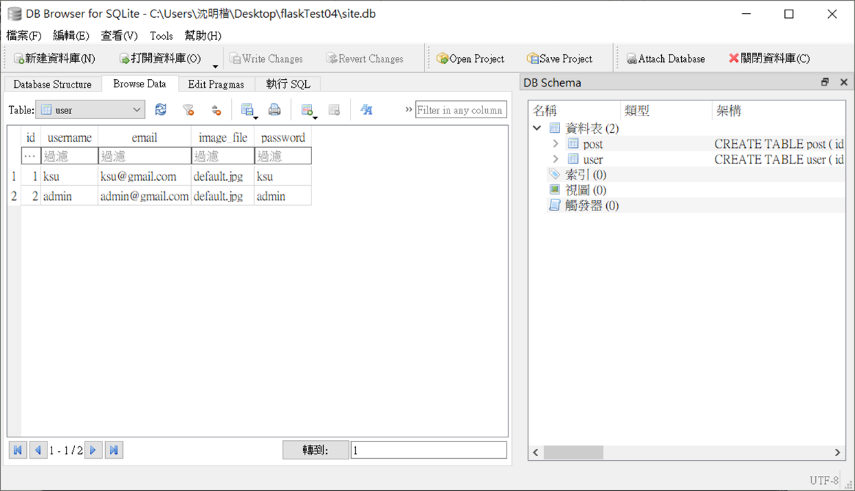
DB Browser for SQLite是一種圖形化管理資料庫工具（如圖4.11），可以設定資料庫欄位，本專題透過此軟體來管理SQLite資料庫，例如：醫師的id及密碼、醫師上傳的X光片（x-ray）以BLOB（Binary Large Object）的格式儲存在資料庫、上傳時間或是辨識結果……等。

圖4. 9 DB Browser軟體示意圖

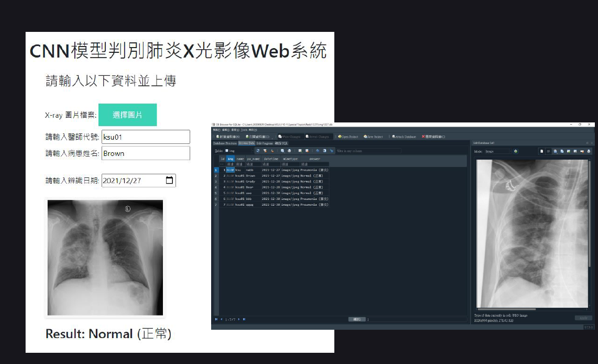
BLOB（Binary Large Object）[16]用來儲存二進位檔案，像是影音、圖片或是大型文件（如圖4.12），本專題利用BLOB格式儲存醫師從前端所上傳的X光片影像。

圖4. 10由本系統（左）上傳資料到資料庫（右）示意圖

4.2 醫學影像訓練子系統

4.2.1 醫學影像蒐集

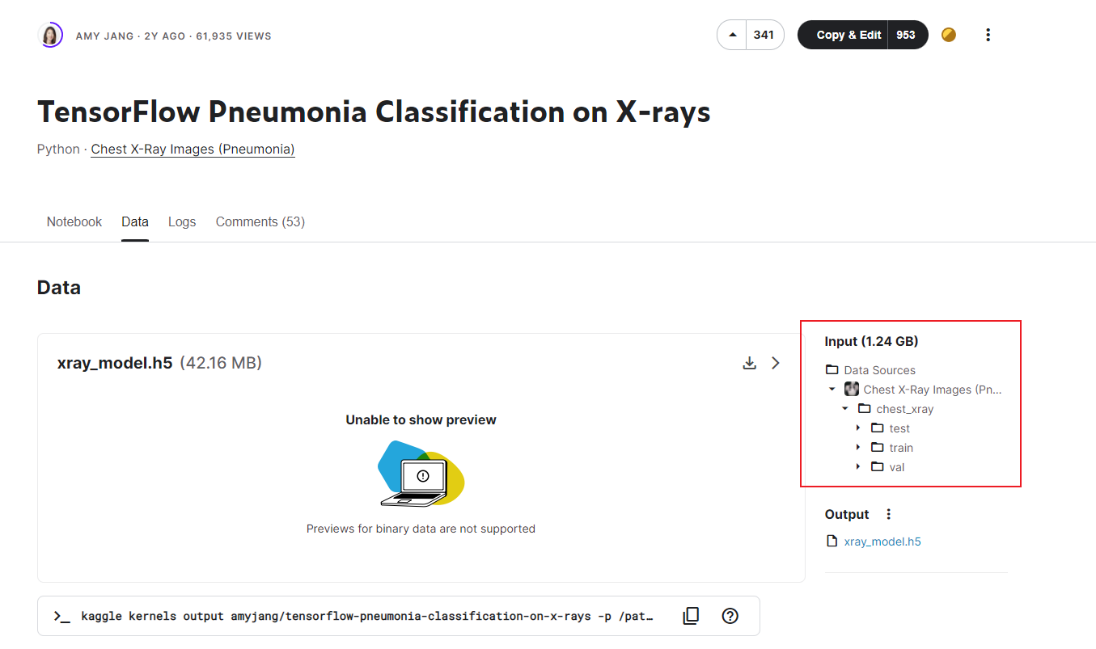
資料集到開放式網站資源Kaggle（如圖4.13）[17]下載到本地端（如圖4.14）。下載後進行影像處理並開始訓練模型，以下為流程與數據圖：

圖4. 11 Kaggle Open Data頁面

圖4. 12將Kaggle數據下載到本地電腦

資料集內有「正常肺部」與「罹患肺炎」兩種影像數據，分別建立資料夾，以備訓練。（如圖4.15、圖4.16）

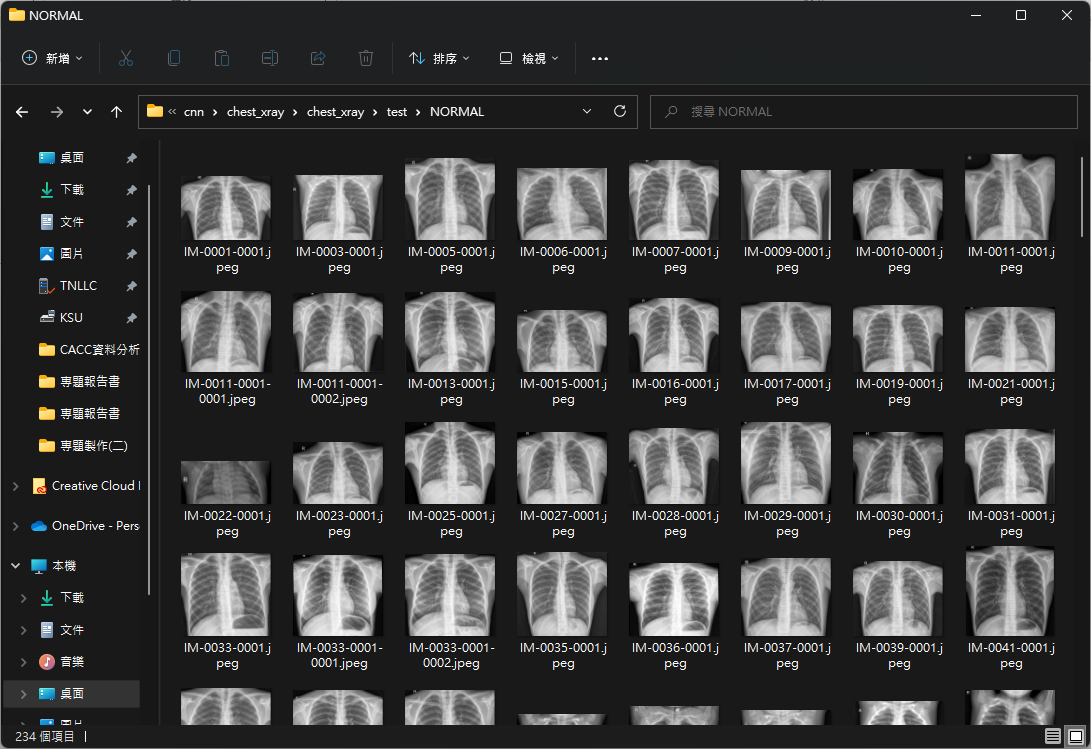
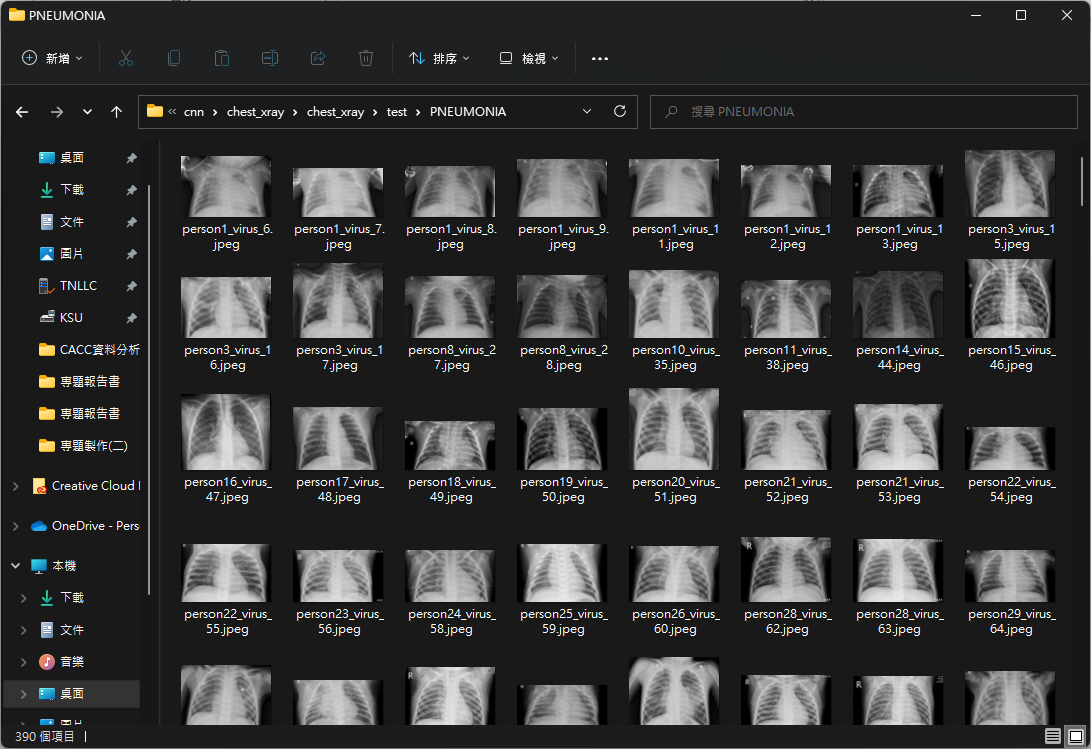
圖4. 13測試集資料（正常肺部）

圖4. 14測試集資料（罹患肺炎）

4.2.2 醫學影像模型訓練

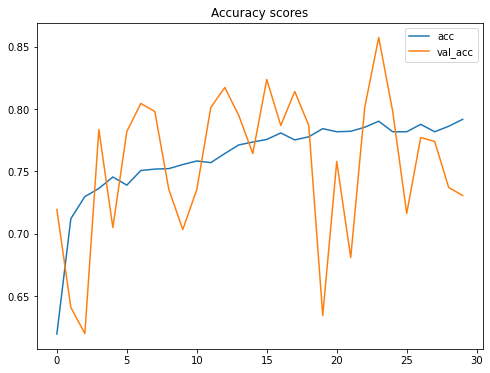
本專題參考Analytics Vidhya平台上所提供之Pneumonia Detection using CNN with Implementation in Python進行模型訓練與測試[18]，整體訓練了五十次，增益率（如圖4.17）可達近八成，遺失率（如圖4.18）低於一成，而訓練後會產生一模型檔案（如圖4.19）。以下是我們訓練後的結果。

圖4. 15影像訓練模型增益率（acc為模型訓練精度、val\_acc為驗證精度）

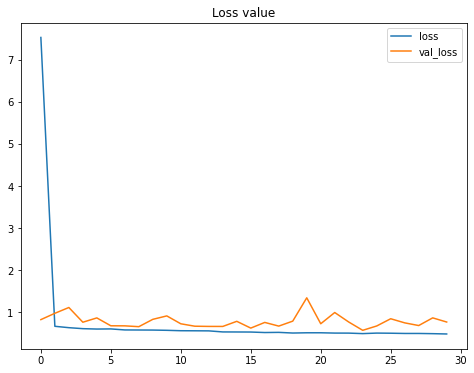
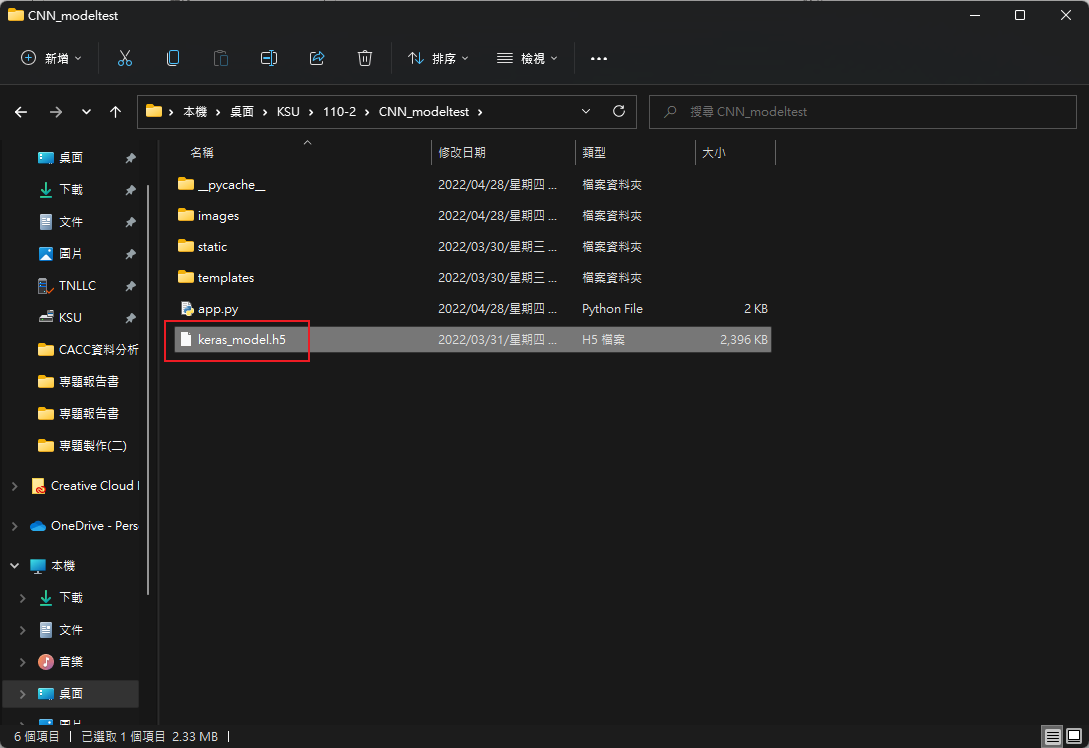


圖4. 16影像訓練模型遺失率（loss為模型訓練損失值、val\_loss為測試損失）

圖4. 17模型訓練後產生之模型檔案（提供給伺服網頁使用）

4.3 使用者Web介面子系統

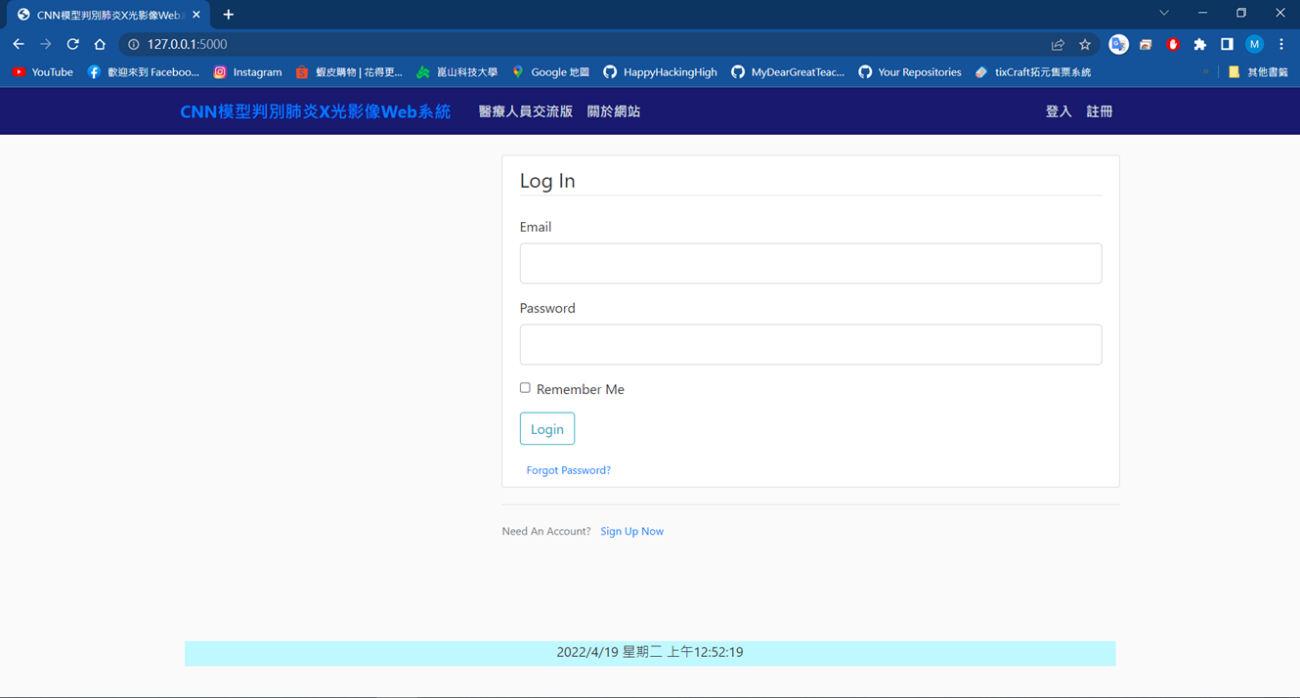
使用者（醫師）登入介面（如圖4.20），讓使用者有專屬的帳號密碼，提供安全性。這個網站必須要由相關醫師才能進入，而非開放式的網站，讓病患的醫療資訊曝光。

圖4. 18使用者前端登入介面

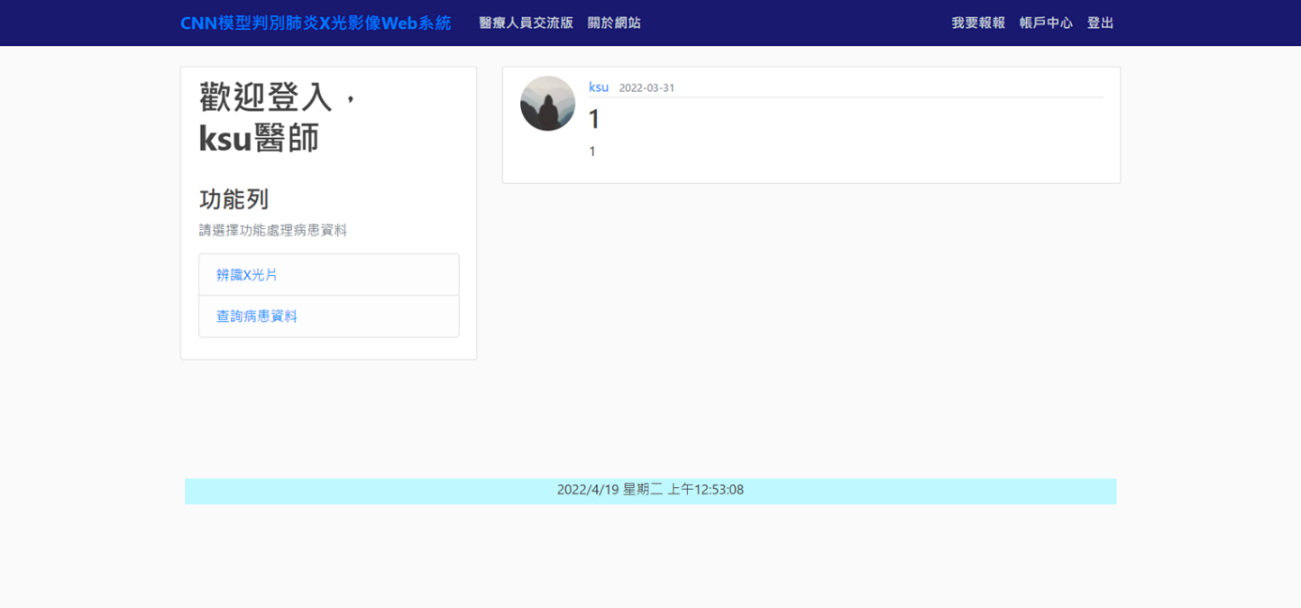
使用者（醫師）登入後的介面，紅框處為該帳密使用者名稱會自動帶入（如圖4.21），例如:登入的使用者名稱為ksu，系統會顯示歡迎登入ksu醫師而不會顯示其他人；若登入的使用者為ksu1，則顯示歡迎登入ksu1醫師；ksu跟ksu1是不同使用者，故無法與用對方的身分去進行其他進一步的操作。

圖4. 19登入後會依帳號顯示醫生名稱於頁面

4.4 後端Web Server子系統

使用者（醫師）點擊sidebar功能列的辨識X光片來上傳患者肺部影像X光透過Web前端介面上傳至Web server（網頁伺服器），X光影像、病患姓名、上傳時間及辨識結果正常（NORMAL）或是肺炎（ PNEUMONIA）會儲存到資料庫（Database）。X光影像是以BLOB的格式來儲存圖片（如圖4.22）。

圖4. 20本專題辨識肺部X光片介面

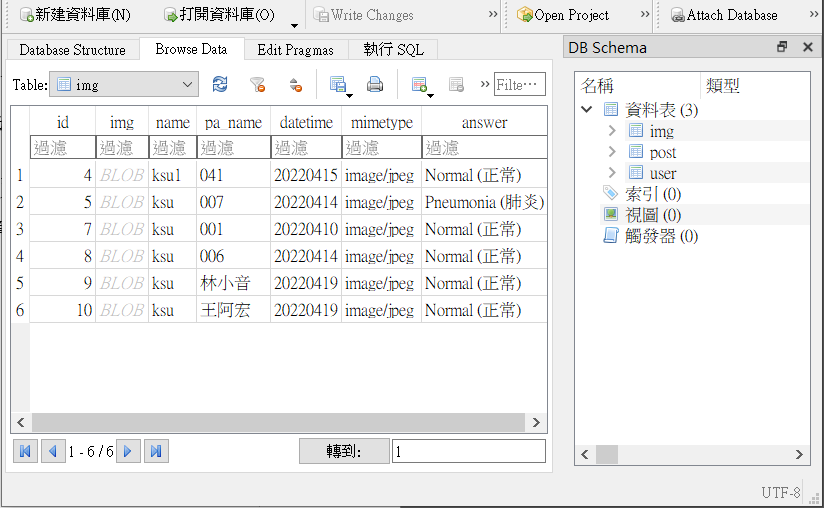
如圖4.23，我們透過DB Browser for SQLite的軟體能查看從辨識X光片介面所上傳的資訊（id、上傳的圖片以BLOB方式儲存在資料庫、上傳圖片的醫生名稱、病患姓名、上傳日期、mimetype辨識上傳檔案的檔名跟辨識結果正常/肺炎）

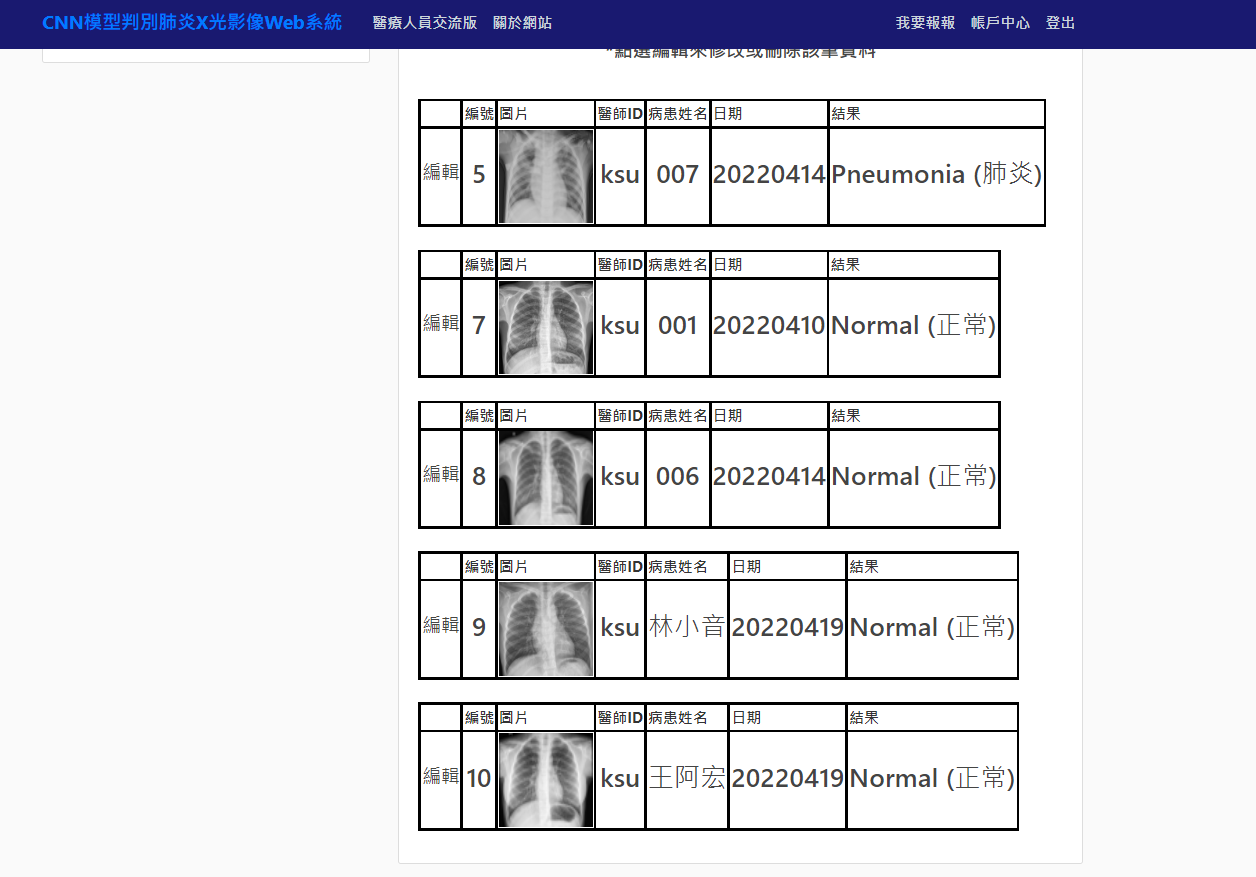
圖4. 21由前端資料上傳至資料庫示意圖

使用者（醫師）可查詢患者肺部影像X光辨識紀錄，但本系統是無法查詢到別人的辨識紀錄，能提供安全性。本系統的輸入框會自動帶入使用者名稱，透過使用者名稱去查詢該使用者的辨識歷史紀錄，使用者無法去更改輸入框，故無法去查詢別人的歷史紀錄來進一步保護隱私。點選查詢按鈕後會列出表格（table）即是辨識歷史紀錄，表格最左邊會有編輯按鈕，下一個步驟會解釋編輯按鈕功能（如圖4.24）。

例如身分為ksu：

圖4. 22本專題辨識記錄查詢介面示意圖

如圖4.25為我們透過名稱為ksu的醫生身分登入，並透過ksu醫生身分去查詢辨識的歷史紀錄，並以列表（table）方式作呈現，而不會藉由ksu醫生身分查詢辨識歷史紀錄會查詢到其他醫師所辨識的紀錄，提供病患資料的安全隱密性。

圖4. 23辨識記錄列表示意圖

若我們是使用ksu1醫師身分去查詢ksu1醫師的辨識歷史紀錄，就只會顯示ksu1這個醫師身分所辨識的所有紀錄，並不會查詢到其他醫師身分所辨識的紀錄。（如圖4.26）

圖4. 24 ksu1身分登入頁面示意圖

使用者（醫師）點擊表格最左邊的編輯會看到Update（更新）與Delete（刪除）的兩個按鈕：

* Update更新按鈕（如圖4.27）：當使用者想要修改錯誤的辨識紀錄，可點選編號左邊編輯的按鈕，即可進入編輯介面，點選Update按鈕就能夠修改病患姓名或上傳影像的日期。若成功刪除會有Flash message來提醒使用者該筆資料已更新。

例如欲更改編號8（ID 8）的資訊：

圖4. 25編輯該筆資料頁面

點擊上圖的Update修改按鈕，會進入到圖4.28修改介面，但是修改只能修改病患姓名與X光片上傳的日期，本系統是以登入身分來決定上傳的醫師是誰，所以在修改的介面不能更改是哪位醫師身分所上傳的，這樣資料庫會亂掉，故只能更改病患姓名與上傳時間（有可能在輸入錯誤需要做更改）。

圖4. 28修改病患姓名示意圖

我們在修改的介面如果填寫完欲更改的文字後，點選儲存按鈕，即可完成修改，這時我們再回到DB Browser for SQLite的軟體，重新整理，會發現我們剛剛更改過的病患姓名及上傳日期已經做更改，Web前端也會顯示flash message提醒使用者紀錄已更新，就表示修改紀錄成功（如圖4.29）。



圖4. 29 由前端修改成功後資料庫更新示意圖

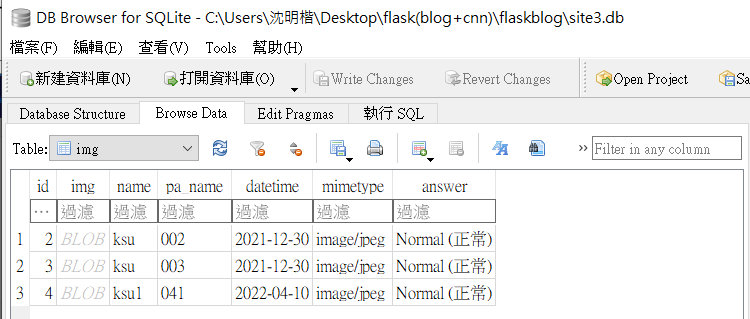
* Delete刪除按鈕：使用者想要刪除某一筆辨識的歷史紀錄（如圖4.30），可點選編號左邊編輯的按鈕，即可進步詢問是否要刪除該筆資料，點擊「是，我要刪除｣字樣即可刪除該筆的紀錄。資料庫也會刪除掉紀錄。

圖4. 26刪除歷史紀錄資料庫示意圖

進入辨識歷史紀錄查詢，點選編輯字樣，點擊Delete刪除按鈕，系統會自動跳出Bootstrap的Model components模態框覆蓋在父表單上的子表單，來進一步提醒與確認使用者是否真的想刪除掉該筆紀錄，會顯示取消及是我要刪除兩個按鈕讓使用者決定是否要刪除該筆資料。（如圖4.31）

圖4. 27系統alert警告是否刪除該筆資料

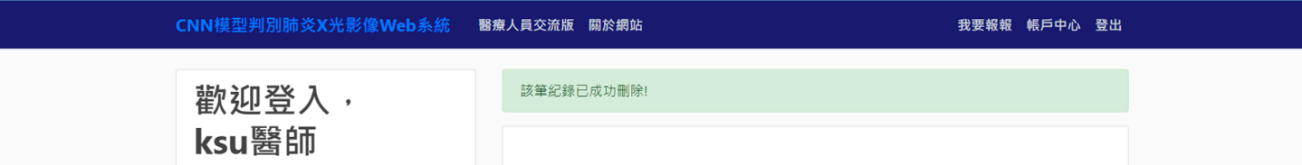
若點選是我要刪除的按鈕，即可成功刪除掉此筆資料，並使用flash message讓使用者知道資料是否有成功被刪除掉。（如圖4.32）

圖4. 28 Flash message顯示該筆紀錄已刪除

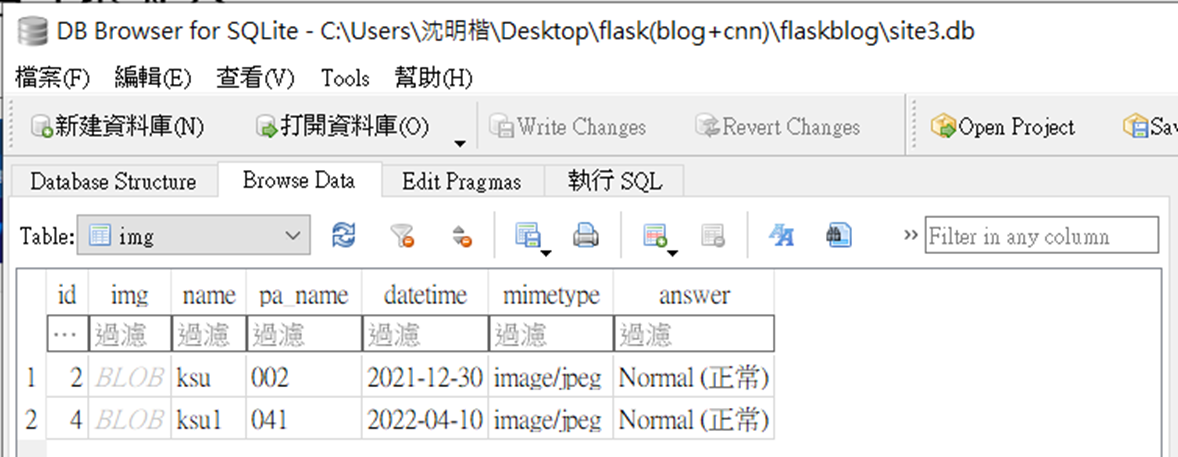
回到DB Browser for SQLite軟體，可以發現剛剛欲刪除的該筆資料已經從資料庫中刪除，表示我們刪除的功能是可以使用的。（如圖4.33）

圖4. 29該筆資料已被刪除示意圖

我們再重新進入ksu醫師的辨識紀錄查詢介面，點選查詢按鈕，即可發現該筆刪除紀錄已經將它刪除，沒有呈現在網頁的table表格內。（如圖4.34）

圖4. 34重新進入查詢也發現該筆紀錄已刪除

第五章 結論

本作品採用CNN （Convolutional Neural Network, 卷積神經網路）模型中的類神經網路演算法（Artificial neural network），將兩種肺炎X光進行模型訓練，再以Web Server將結果顯現在Web介面（當然Web介面中會設有讓特定醫師的權限），希望藉由人工智慧結合醫療，透過AI輔助醫療來協助醫護人員，分擔醫護人員心力，也能將此系統結合X光車，讓在偏鄉的居民，也能方便的去做更精密的X光掃描，不用辛勞跑到市區醫院做檢查，可直接拍攝胸腔X光後，藉由系統快速告知醫師結果。

在未來希望能為了新冠肺炎疫情做出貢獻，這場疫情至今仍嚴峻，醫療量能也很緊繃，醫護人員休息的時間也有限，若我們將系統做些改良，就能協助分擔醫護人員心力，我們期盼透過科技減輕醫護人員之辛勞，試將此想法以人工智慧結合醫療，分擔醫護人員工作量能，患者就能夠及早治療。

第六章 參考文獻

1. Covid-19全球疫情地圖，<https://covid-19.nchc.org.tw/dt_005-covidTable_7day_confirmed.php>
2. Convolutional Neural Networks （CNNs / ConvNets），<https://cs231n.github.io/convolutional-networks/>
3. 深度學習網路學習筆記（一）-全連接層，旅人\_Eric，<https://blog.csdn.net/qq_37099552/article/details/100918664>，2019-09-17
4. 循環神經網路，百科知識，<https://www.easyatm.com.tw/wiki/%E5%BE%AA%E7%92%B0%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF>
5. 卷積神經網路 （Convolutional Neural , CNN），李宏毅，<https://hackmd.io/@allen108108/rkn-oVGA4>
6. A Gentle Introduction to Pooling Layers for Convolutional Neural Networks，Jason Brownlee，<https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>，2019/04/22
7. Average Pooling，<https://paperswithcode.com/method/average-pooling>
8. Simple integration of Flask and WTForms, including CSRF, file upload, and reCAPTCHA.，<https://flask-wtf.readthedocs.io/en/1.0.x/>
9. Flask，維基百科，<https://zh.m.wikipedia.org/zh-tw/Flask>
10. Flask-Routing，[https://flask.palletsprojects.com/en/2.1.x/quickstart/#routing](https://flask.palletsprojects.com/en/2.1.x/quickstart/%23routing)
11. Flask HTTP methods, handle GET & POST requests，Python Tutorial，<https://pythonbasics.org/flask-http-methods/>
12. Flask-Template，[https://flask.palletsprojects.com/en/2.1.x/quickstart/#rendering-templates](https://flask.palletsprojects.com/en/2.1.x/quickstart/%23rendering-templates)
13. Flask實作\_基礎\_10\_flash message，<https://hackmd.io/@shaoeChen/BkkbvYIff?type=view>
14. Child Template，<https://flask.palletsprojects.com/en/1.1.x/patterns/templateinheritance/>
15. Flask-SQLAlchemy，<https://flask-sqlalchemy.palletsprojects.com/en/2.x/>
16. Binary Large Objects，<https://docs.oracle.com/cd/E17276_01/html/api_reference/C/blob.html>
17. TensorFlow Pneumonia Classification on X-rays，<https://www.kaggle.com/code/amyjang/tensorflow-pneumonia-classification-on-x-rays/data>
18. Pneumonia Detection using CNN with Implementation in Python，<https://www.analyticsvidhya.com/blog/2020/09/pneumonia-detection-using-cnn-with-implementation-in-python/>，2020/09/16