

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：陳冠曉

學號：4070E020

專題組員 3：沈明楷


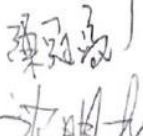
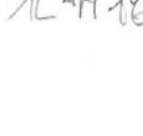
學號：4070E022

中華民國 111 年 05 月

崑山科技大學  
資訊工程系  
110學年度專題製作報告

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.

學	生	
<u>蘇宇祥</u>	<u>4070E021</u>	
<u>陳冠曉</u>	<u>4070E020</u>	
<u>沈明楷</u>	<u>4070E022</u>	

指導老師：李宇倫 (簽章)

# CNN 模型判別肺炎 X 光影像 Web 平台系統

蘇宇祥 陳冠曉 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，也誨人不倦的耐心指導與協助，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
Abstract .....	ii
第一章 緒 論 .....	1
1.1 研究動機與目的 .....	1
1.2 研究方法.....	1
1.2.1 全連接層（Full Connected layer） .....	1
1.2.2 卷積層（Convolution layer） .....	2
1.2.3 池化層（Pooling layer） .....	2
1.2.4 展平層（Flatten layer） .....	4
1.2.5 正規化 Softmax 函式 .....	4
1.3 系統概述.....	5
1.4 工作分配.....	5
第二章 系統相關技術介紹 .....	6
2.1 Python .....	6
2.2 Flask.....	6
2.2.1 Flask 簡介 .....	6
2.2.2 Route 路由.....	6
2.2.3 HTTP Methods（GET & POST requests） .....	6
第三章 系統流程與架構.....	8
3.1 系統操作流程.....	8
3.2 系統架構.....	8
3.3 子系統.....	9
3.3 資料庫結構.....	10
第四章 系統功能說明.....	11
4.1 系統語法說明 .....	11
4.1.1 url_for（）路由設定.....	11
4.1.2 flash message 訊息顯示.....	11
4.1.3 extends 模板繼承 .....	13
4.1.4 Flask-SQLAlchemy 資料庫語法 .....	15
4.1.5 SQLite 管理軟體 .....	16
4.2 醫學影像訓練子系統 .....	17
4.2.1 醫學影像蒐集 .....	17
4.2.2 醫學影像模型訓練.....	19
4.3 使用者 Web 介面子系統 .....	21
4.4 後端 Web Server 子系統.....	21
第五章 結論 .....	28

第六章 參考文獻 ..... 29

## 圖目錄

圖 1. 1 卷積層運算過程.....	2
圖 1. 2 最大池化層示意圖.....	4
圖 1. 3 平均池化層示意圖.....	4
圖 1. 4 展平層示意圖.....	4
圖 2. 1 Flask pip 安裝套件語法示意圖.....	6
圖 3. 1 本組系統使用者頁面.....	8
圖 3. 2 上傳資料進入資料庫.....	8
圖 3. 3 本專題系統架構圖.....	9
圖 3. 4 本專題三大子系統架構.....	9
圖 3. 5 本專題 ERD 資料庫關係圖.....	10
圖 4. 1 url_for ( ) 語法示意圖.....	11
圖 4. 2 import Flash 示意圖.....	12
圖 4. 4 將迴圈程式新增到欲顯示的 html.....	12
圖 4. 5 前端系統刪除介面的 flash message.....	12
圖 4. 6 繼承語法說明圖.....	13
圖 4. 7 本專題 home.html 父模板.....	13
圖 4. 8 本專題 upload.html 子模板.....	14
圖 4. 10 Flask-SQLAlchemy.....	15
圖 4. 11 DB Browser 軟體示意圖.....	16
圖 4. 12 由本系統 ( 左 ) 上傳資料到資料庫 ( 右 ) 示意圖.....	16
圖 4. 13 Kaggle Open Data 頁面[16].....	17
圖 4. 14 將 Kaggle 數據下載到本地電腦.....	17
圖 4. 15 測試集資料 ( 正常肺部 ).....	18
圖 4. 16 測試集資料 ( 罹患肺炎 ).....	18
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
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
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
圖 4. 20 使用者前端登入介面.....	21
圖 4. 21 登入後會依帳號顯示醫生名稱於頁面.....	21
圖 4. 22 本專題辨識肺部 X 光片介面.....	22
圖 4. 23 由前端資料上傳至資料庫示意圖.....	22
圖 4. 24 本專題辨識記錄查詢介面示意圖.....	22
圖 4. 25 辨識記錄列表示意圖.....	23
圖 4. 26 ksul 身分登入頁面示意圖.....	23
圖 4. 27 編輯該筆資料頁面.....	24
圖 4. 30 刪除歷史紀錄資料庫示意圖.....	25



圖 4.31 系統 alert 警告是否刪除該筆資料.....	26
圖 4.32 Flash message 顯示該筆紀錄已刪除 .....	26
圖 4.33 該筆資料已被刪除示意圖.....	27

## 表目錄

表 1.1 工作分配表.....	5
表 4.1 資料庫示意表.....	15

# 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，每個特徵面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  $\otimes$ ）（如

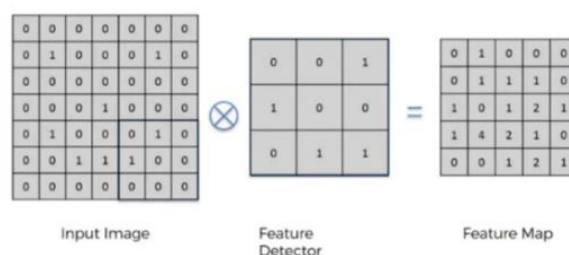


圖 1.1)。最常用於訊號分析與影像處理。

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，在眾多湧快數據中，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）



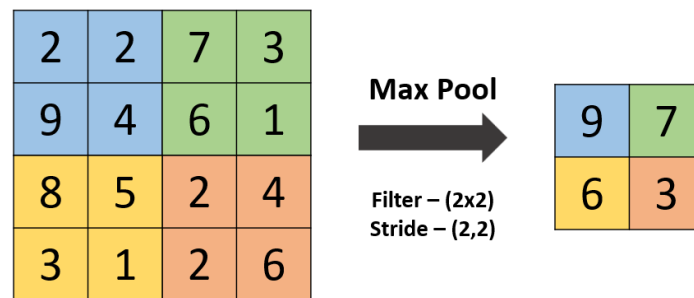


圖 1.2 最大池化層示意圖

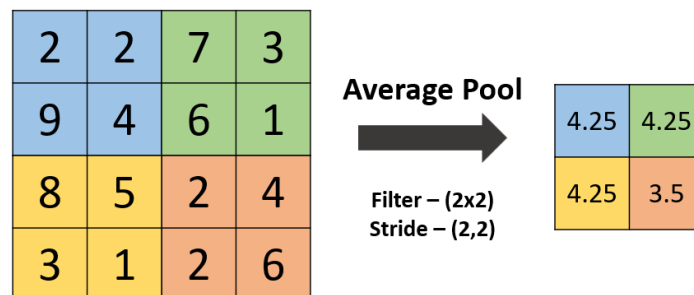


圖 1.3 平均池化層示意圖

#### 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）。

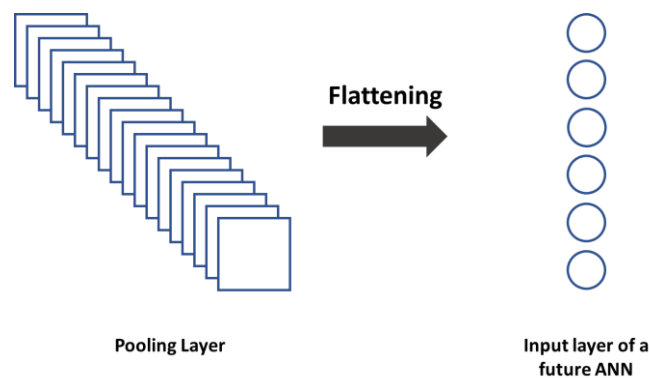


圖 1.4 展平層示意圖

#### 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器 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端上傳，影像透過與置於伺服器子系統之模型比對，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。
- 網頁伺服器 Web Server 子系統：利用 Python Flask 撰寫伺服器網頁程式，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
陳冠曉	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）。



```
pip install flask
```

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器網頁之模型比對後，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

	id	img	name	pa name	datetime	mimetype	answer
	過濾	過濾	過濾	過濾	過濾	過濾	過濾
1	4	<i>BLOB</i>	ksu1	041	20220415	image/jpeg	Normal (正常)
2	5	<i>BLOB</i>	ksu	007	20220414	image/jpeg	Pneumonia (肺炎)
3	7	<i>BLOB</i>	ksu	001	20220410	image/jpeg	Normal (正常)
4	8	<i>BLOB</i>	ksu	006	20220414	image/jpeg	Normal (正常)
5	9	<i>BLOB</i>	ksu	林小音	20220419	image/jpeg	Normal (正常)
6	10	<i>BLOB</i>	ksu	王阿宏	20220419	image/jpeg	Normal (正常)

資料表 (3)

>  img

>  post

>  user

 索引 (0)

 視圖 (0)

 觸發器 (0)

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)

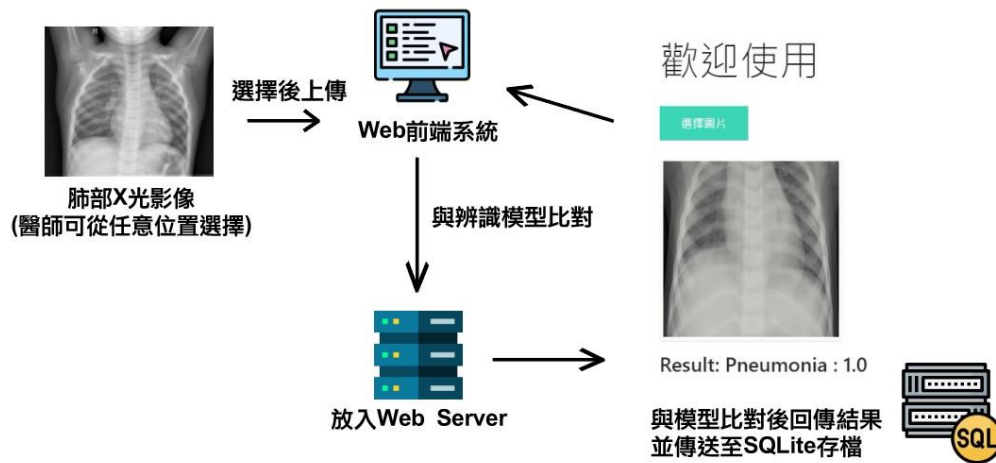


圖 3.3 本專題系統架構圖

### 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)

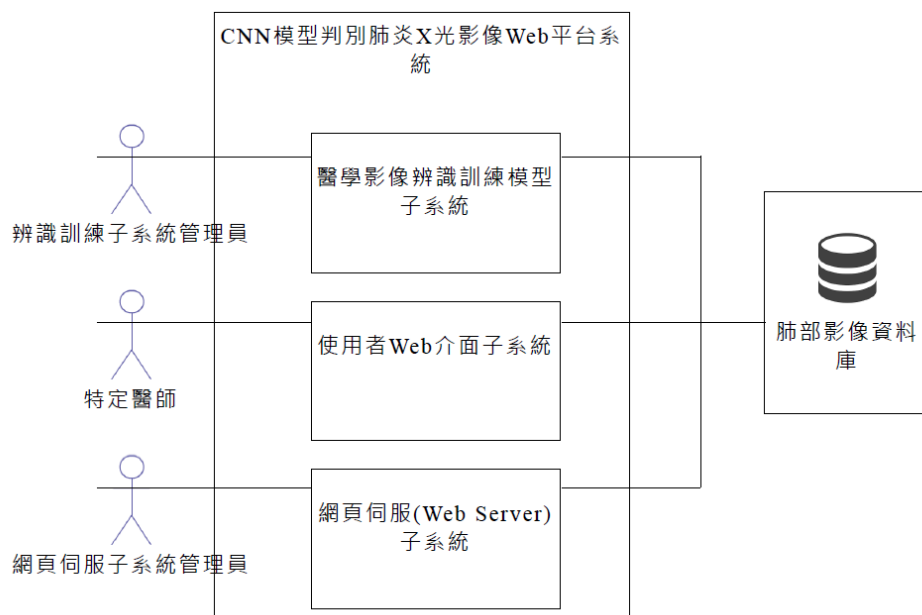


圖 3.4 本專題三大子系統架構

### 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)

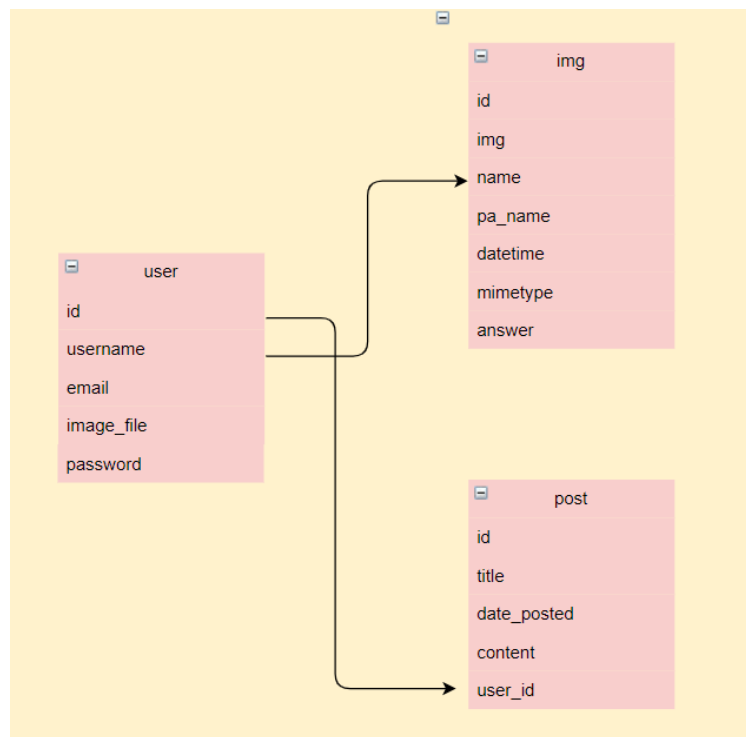


圖 3.5 本專題 ERD 資料庫關係圖

## 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。

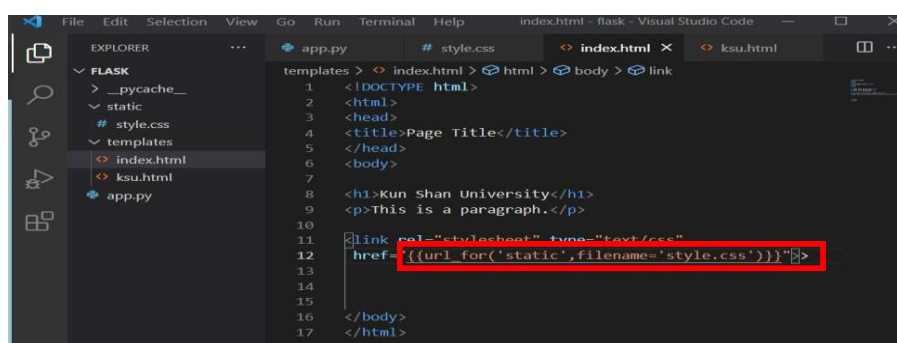


圖 4.1 url\_for ( ) 語法示意圖

#### 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。

```
app.py > ...  
1 from __future__ import division, print_function  
2 from re import A  
3 from flask import Flask, render_template, request, flash
```

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。

```
app = Flask(__name__)  
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///img1227.db'  
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False  
app.config['SECRET_KEY'] = 'cnn'  
  
db_init(app)
```

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）。

```
{% for message in get_flashed_messages(with_categories=True) %}  
<div class="alert alert-{{ message[0] }}">  
    <button type="button" class="close" data-dismiss="alert">&times;</button>  
    {{ message[1] }}  
</div>  
{% endfor %}
```

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。

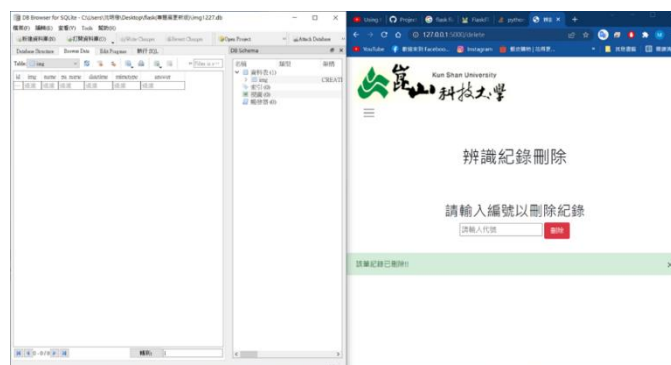


圖 4.4 前端系統刪除介面的 flash message

## 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）。

"bootstrap/base.html" 就是 Flask-Bootstrap 自帶的基模板，我們自己的基模板也要繼承它。

```
{% block head %}
{% endblock %}
```

這個是 Jinja2 的自定義塊，在 Flask-Bootstrap 中有很多定義好的塊：

塊名	說明
doc	整個 HTML 文件
html_attribs	<html> 標籤中的屬性
html	<html> 標籤中的內容
head	<head> 標籤中的內容
title	<title> 標籤中的內容
metas	一組 <meta> 標籤
styles	CSS 定義
body_attribs	<body> 標籤中的屬性
body	<body> 標籤中的內容
navbar	使用自定義的導航欄
content	使用自定義的頁面內容
scripts	文件底部的 JS 腳本

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 (`extends`) 的導覽列 `navbar`，`home.html` 的 `{% 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





#### 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]

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///tmp/test.db'
db = SQLAlchemy(app)

class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)

    def __repr__(self):
        return '<User %r>' % self.username
```

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。

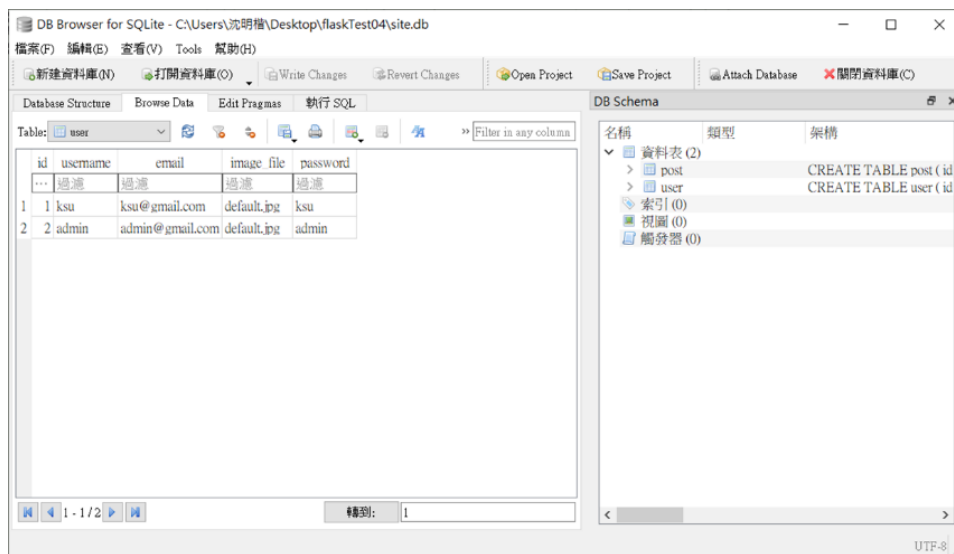


圖 4.9 DB Browser 軟體示意圖

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。

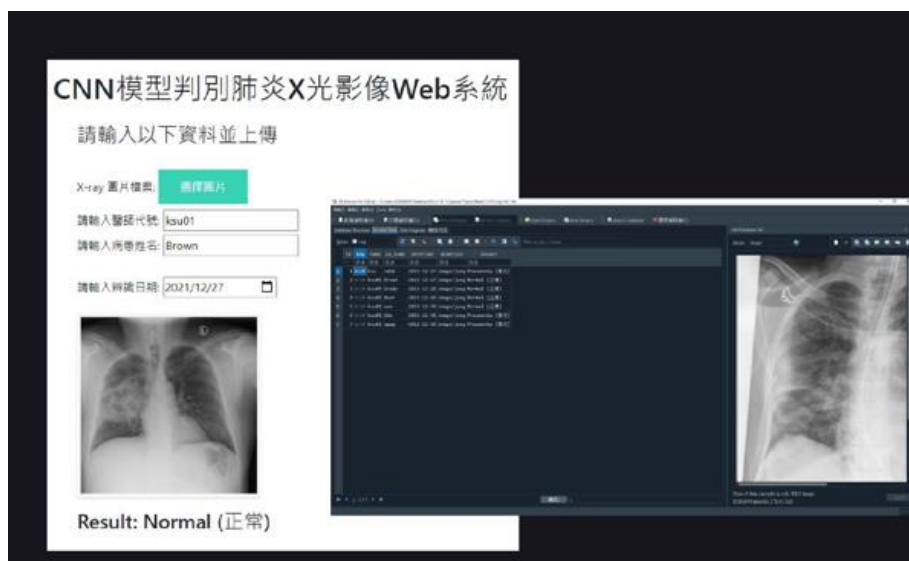


圖 4.10 由本系統（左）上傳資料到資料庫（右）示意圖

## 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：

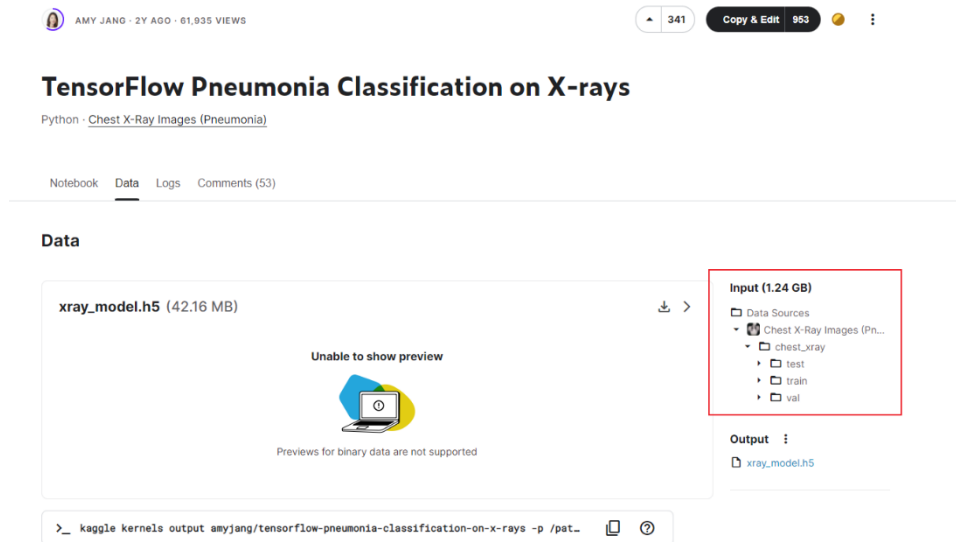


圖 4.11 Kaggle Open Data 頁面

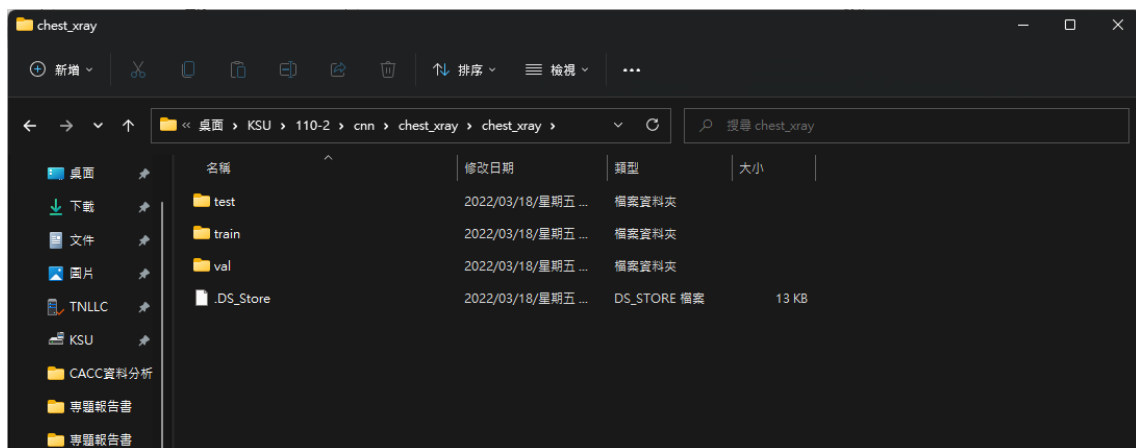


圖 4.12 將 Kaggle 數據下載到本地電腦

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)

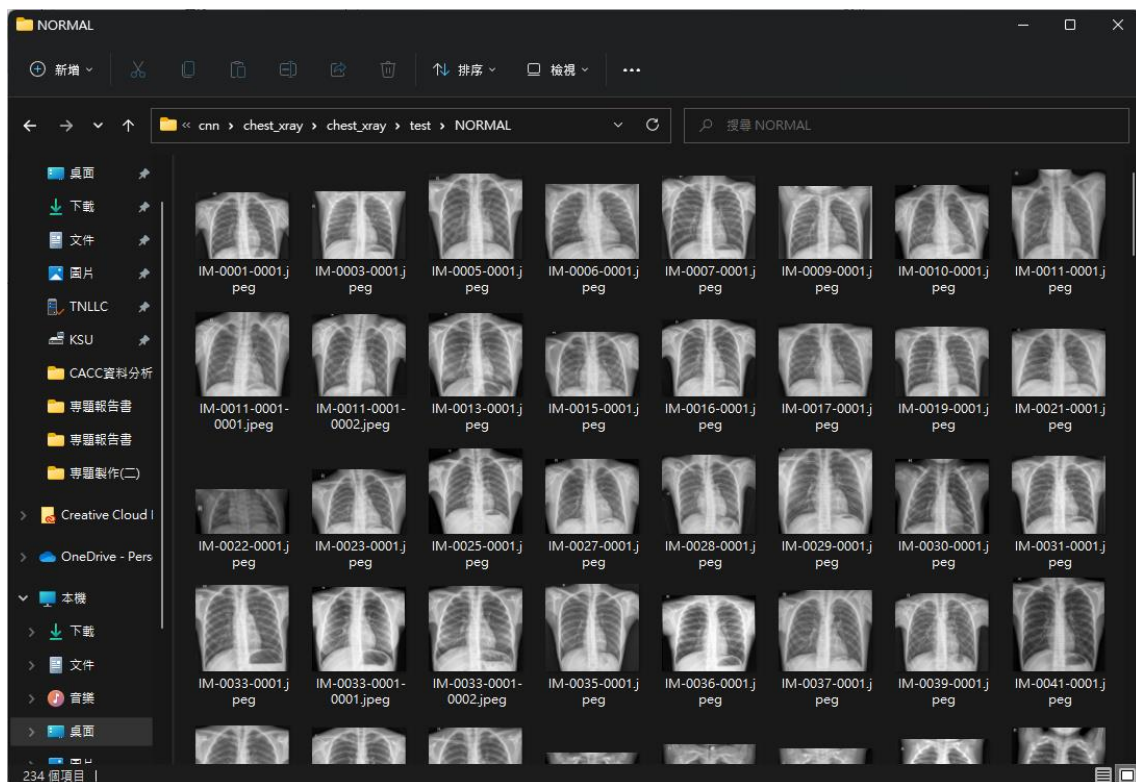


圖 4.13 測試集資料（正常肺部）

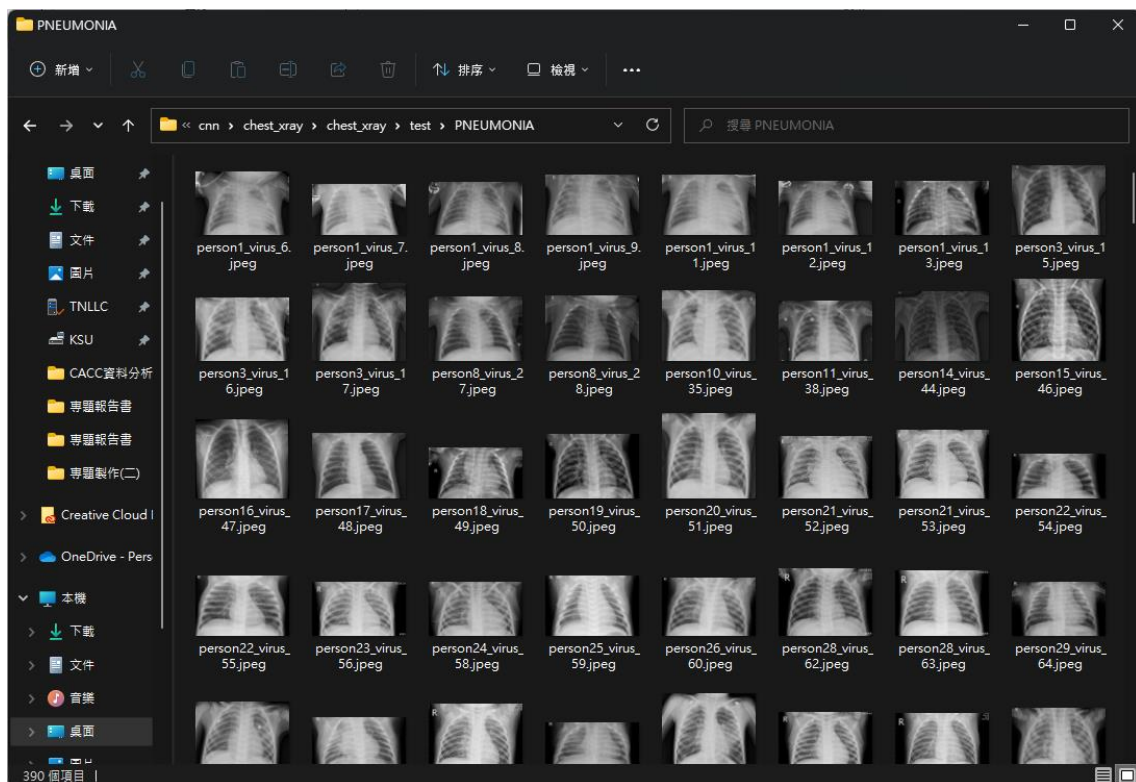


圖 4.14 測試集資料（罹患肺炎）

### 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。

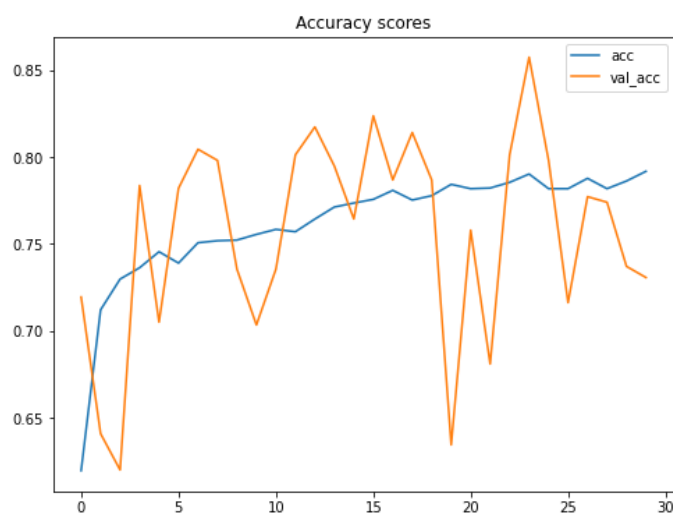


圖 4.15 影像訓練模型增益率（acc 為模型訓練精度、val\_acc 為驗證精度）

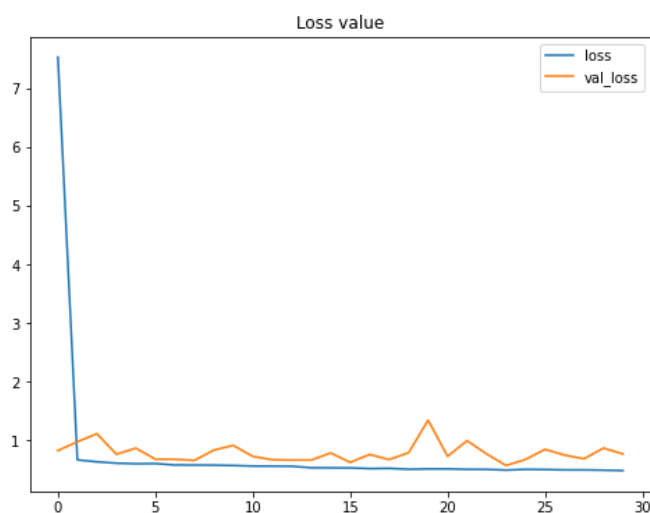


圖 4.16 影像訓練模型遺失率（loss 為模型訓練損失值、val\_loss 為測試損失）

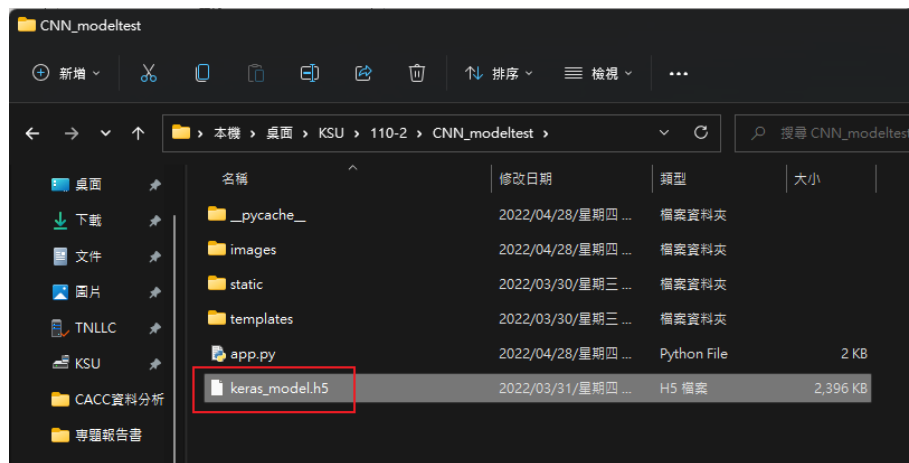


圖 4. 17 模型訓練後產生之模型檔案（提供給伺服網頁使用）

### 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。

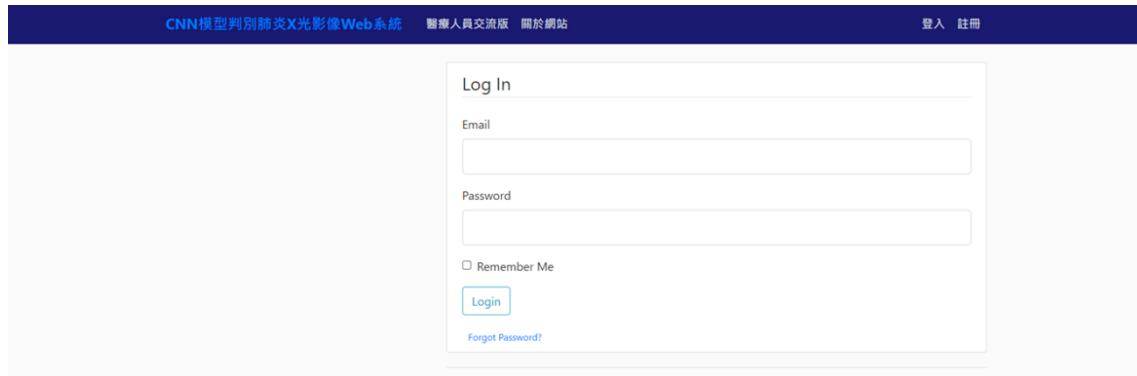


圖 4.18 使用者前端登入介面

使用者（醫師）登入後的介面，紅框處為該帳號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）。



**歡迎登入，ksu醫師**

**功能列**  
請選擇功能處理病患資料

辨識X光片

查詢病患資料


### 肺炎辨識

X-ray 圖片檔案:  
選擇圖片

醫師名稱(不能修改)

請輸入病患姓名:

請輸入辨識日期:(YYYYMMDD)  
Ex: 19990904



**Result: Normal (正常)**

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、上傳的图片以 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）

	id	img	name	pa_name	datetime	mimetype	answer
	過濾	過濾	過濾	過濾	過濾	過濾	過濾
1	4	BLOB	ksu1	041	20220415	image/jpeg	Normal (正常)
2	5	BLOB	ksu	007	20220414	image/jpeg	Pneumonia (肺炎)
3	7	BLOB	ksu	001	20220410	image/jpeg	Normal (正常)
4	8	BLOB	ksu	006	20220414	image/jpeg	Normal (正常)
5	9	BLOB	ksu	林小音	20220419	image/jpeg	Normal (正常)
6	10	BLOB	ksu	王阿宏	20220419	image/jpeg	Normal (正常)

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：

**歡迎登入，ksu醫師**

**功能列**  
請選擇功能處理病患資料

辨識X光片

查詢病患資料

### 辨識紀錄查詢

查詢

**步驟1. 點擊查詢按鈕以繼續**

\*點選編輯來修改或刪除該筆資料

圖 4.22 本專題辨識紀錄查詢介面示意圖



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。






CNN模型判別肺炎X光影像Web系統 醫療人員交流版 關於網站 我要報報 帳戶中心 登出					
點選編輯不修改或刪除改單資料					
編號	圖片	醫師ID	病患姓名	日期	結果
5		ksu	007	20220414	Pneumonia (肺炎)
編號	圖片	醫師ID	病患姓名	日期	結果
7		ksu	001	20220410	Normal (正常)
編號	圖片	醫師ID	病患姓名	日期	結果
8		ksu	006	20220414	Normal (正常)
編號	圖片	醫師ID	病患姓名	日期	結果
9		ksu	林小音	20220419	Normal (正常)
編號	圖片	醫師ID	病患姓名	日期	結果
10		ksu	王阿宏	20220419	Normal (正常)

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)

<div> <div>歡迎登入，ksu1醫師</div> <div>功能列</div> <div>請選擇功能處理病患資料</div> <div> <div>辨識X光片</div> <div>查詢病患資料</div> <div>刪除資料</div> <div>修改資料</div> </div> </div>						
<div> <div>辨識紀錄查詢</div> <div>點擊查詢按鈕以繼續(不能修改)</div> <div> <div>ksu1</div> <div>查詢</div> </div> </div>						
更多...	4		ksu1	041	2022-04-10	Normal (正常)

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)

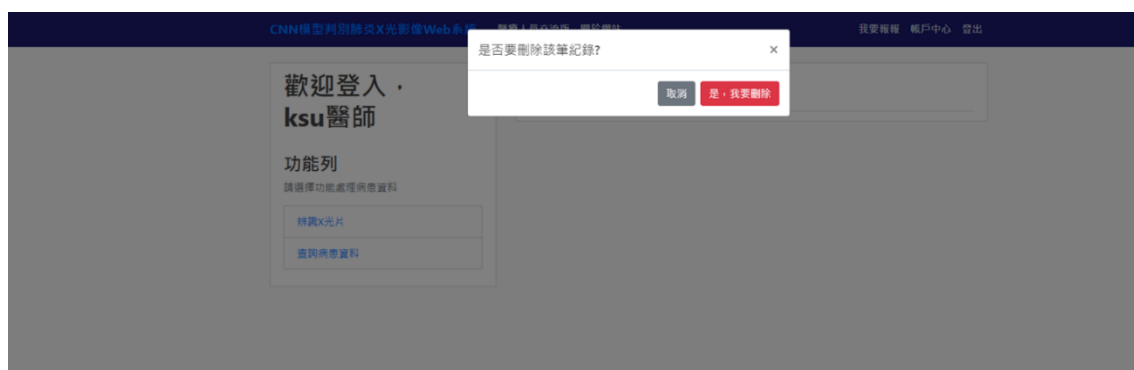


圖 4.27 系統 alert 警告是否刪除該筆資料

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)

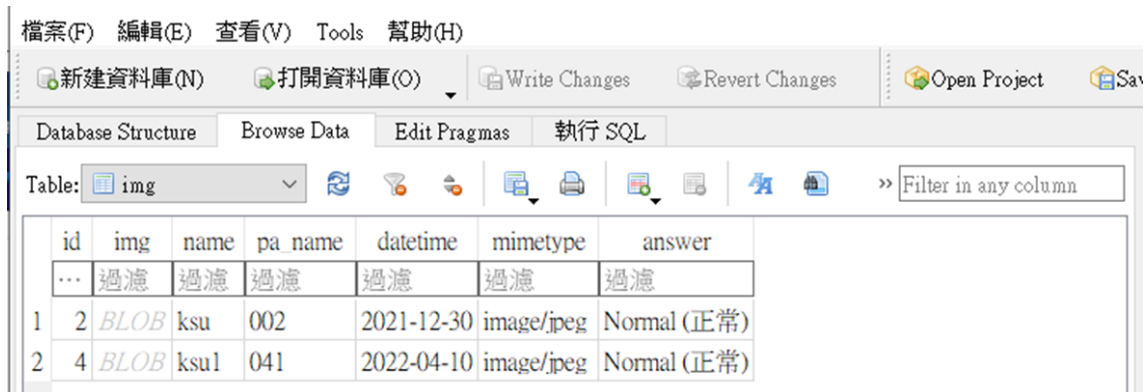


圖 4.29 該筆資料已被刪除示意圖

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](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](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>
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>
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](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