

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

**A Web Platform System of CNN Model
Discriminate Lungs' X-ray Image**

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
學生專題製作報告

CNN 模型判別肺炎 X 光影像 Web 平台系統

**A Web Platform System of CNN Model
Discriminate Lungs' X-ray Image**

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

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

新型冠狀肺炎 (Covid-19, Corona Virus Disease 2019) 近年來肆虐全球，截至 2022 年 3 月 27 日，全球已累計報告逾 4.80 億例確診個案，其中逾 612.1 萬人死亡，病死率約為 1.27%，是人類歷史上大規模流行病之一。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，因此當醫護人員檢測時，除了進行抗原快篩與核酸檢測 (PCR) 外，醫師會透過照胸腔 X 光來查看病情，不過隨著病例增加，以及利用傳統的人工判讀肺部 X 光影像檢驗方式，醫師查看相關資料時可能會有誤判情況發生，可能導致患者失去黃金醫療時間，為此希望能夠解決判讀上的問題。

而現代科技日新月異，科技開始能介入人類生活，能解決許多問題，漸漸地人工智慧 (Artificial Intelligence, AI) 與影像辨識技術開始與醫療結合，渴望透過科技解決當前所遇問題，達到 AI 生物醫療，希望可以藉由人工智慧協助，讓醫師有更精確的判斷，患者能夠及早治療、對症下藥，把握黃金時間。

關鍵詞：新型冠狀肺炎、影像辨識、生物醫療

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	3
第一章 緒 論	10
1.1 研究動機與目的	10
1.2 研究方法	10
1.2.1 卷積神經網路 (CNNs/ ConvNets)	10
1. 全連接網路	10
1.1 輸入層	11
1.2 隱藏層	11
1.3 輸出層	11
2. 卷積神經網路	11
2.1 卷積層	11
2.2 池化層	12
2.2.1 Max pooling (最大池化)	12
2.2.2 Average pooling (平均池化) [6]	12
1.3 系統概述	13
1.4 工作分配	13
第二章 系統相關技術介紹	14
2.1 Python	14
2.2 Flask	14
2.2.1 Flask 簡介	14
2.2.2 Route 路由	14
2.2.3 HTTP Methods (GET & POST requests)	14
2.2.4 Templates (動態文件)	15
2.2.5 Static (靜態文件)	15
2.2.6 url_for ()、redirect	15
2.2.7 flash message	15
2.2.8 extends (模板繼承)	16
2.3 Flask-SQLAlchemy[13]	18
2.4 SQLite 管理工具	19
第三章 系統流程與架構	20
3.1 系統操作流程	20
3.2 系統架構	21
第四章 系統功能說明	21
4.1 醫學影像訓練子系統	21
4.1.1 醫學影像蒐集	21
4.1.2 醫學影像模型訓練[16]	24

4.2 使用者 Web 介面子系統.....	25
4.3 後端 Web Server 子系統	25
第五章 結論	30
第六章 參考文獻	31

圖目錄

圖 1 卷積層運算過程	11
圖 2 最大池化層示意圖	12
圖 3 平均池化層示意圖	12
圖 4 Flask pip 安裝套件	14
圖 5 url_for () 示意圖	15
圖 6 import Flash.....	16
圖 7 設定 secret key 加密	16
圖 8 將迴圈程式新增到欲顯示的 html.....	16
圖 9 刪除介面的 flash message.....	16
圖 10 繼承語法	17
圖 11 home.html 父模板	17
圖 12 upload.html 子模板	18
圖 13 upload.html 已繼承 home.html 的 navbar 導覽列.....	18
圖 14 Flask-SQLAlchemy.....	18
圖 15 DB Browser 軟體	19
圖 16 醫師上傳 x 光至資料庫	20
圖 17 本組系統使用者頁面	20
圖 18 上傳資料進入資料庫	21
圖 19 系統架構圖	21
圖 20 Kaggle Open Data	22
圖 21 將數據下載到本地電腦	22
圖 22 測試集資料(正常肺部)	23
圖 23 測試集資料(罹患肺炎)	23
圖 24 影像訓練模型增益率	24
圖 25 影像訓練模型遺失率	24
圖 26 模型訓練後產生之模型檔案(提供給伺服網頁使用)	24
圖 27 使用者前端登入介面	25
圖 28 登入後會依照帳號顯示醫生名稱	25
圖 29 辨識 X 光片介面	26
圖 30 資料上傳至資料庫中	26
圖 31 辨識記錄查詢介面	26
圖 32 辨識記錄列表	27
圖 33 ksul 身分登入頁面.....	27
圖 34 編輯頁面畫面，並點選 Update 按鈕，但是只有病患姓名與上傳日期能更改	28
圖 35 修改病患姓名	28

圖 36	Flash message 顯示已更新，資料庫也更新成功	28
圖 37	刪除前之資料庫	29
圖 38	系統 alert 警告是否刪除該筆資料	29
圖 39	Flash message 顯示該筆紀錄已刪除	29
圖 40	該筆資料已被刪除	29
圖 41	重新進入查詢也發現該筆紀錄已刪除	30

表目錄

表 1 組員工作分配表	13
表 2 資料庫示意表	19

第一章 緒 論

1.1 研究動機與目的

近年來 COVID-19 (Covid-19, Corona Virus Disease 2019) 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 生物醫療，希望可以藉由人工智慧協助，讓醫師有更精確的判斷，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。以類神經網路演算法 (Artificial Neural Network, ANN) 為主要演算法。

1.2.1 卷積神經網路 (CNNs/ ConvNets)

卷積神經網路[1]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、卷積神經網路 (Convolutional Neural Network, CNN) 和循環神經網路 (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. 全連接網絡

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[2]，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，把我們經過數個卷積、池化後的結果進行分類。全連接層能將學到的「分散式特徵表示」映射到樣本標記空間的作用由三個部分組成，分別是輸入層 (Input Layers)、

隱藏層（Hidden Layers）以及輸出層（Output Layers）。[3]

1.1 輸入層

輸入層將輸入資料提出特徵值，舉例來說，要預測一個貓還是狗，我們將現有的資料可能是外觀、體型，而這些就是你要放進輸入層的特徵。

1.2 隱藏層

隱藏層就是在背後進行運算，由神經元組成，透過前向傳播（Forward propagation）計算出我們的所輸出的值。

1.3 輸出層

而輸出層透過所輸入特徵值，經過隱藏層的演算所得到的預測值，再藉由縮小預測值與實際標準值，讓我們隱藏層的參數，最後訓練出一組判斷率相對上高的模型。

2. 卷積神經網路

近幾年來，隨著快速發展人工智慧，深度學習受到越來越多的研究者的使用。卷積神經網路，因它的圖像分類、目標檢測、圖像語義分割等領域取得一定的成果，卷積神經網路是由多張圖片中提取特徵的卷基層（convolutional layer）和用於特徵處理的取樣層（pooling layer，也稱為池化層），交疊組成的多層神經網。[4]

2.1 卷積層

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，卷基層是一個矩陣（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 ）

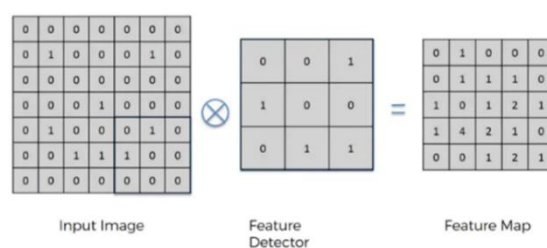


圖 1 卷積層運算過程

中間的 Feature Detector (Filter) 會隨機產生，目的是從中篩選圖片當中的一些特徵，進行了多次卷積後，可能因操作使用不同的過濾器，最終導致特徵顯現的不同，最後將這些特徵圖片放在一起，作為卷積層最後的輸出。

2.2 池化層

池化層[5]主要用於做圖片的特徵萃取，其過程中，圖形的縮放不會影響，經由這樣的 scaling 我們也可以再一次的減少神經網路的參數，優點為縮減模型的大小且提高計算速度。常見的池化方式有 Max pooling (最大池化) 與 Average pooling (平均池化) 兩種。

2.2.1 Max pooling (最大池化)

最大池化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 2.2.1)。

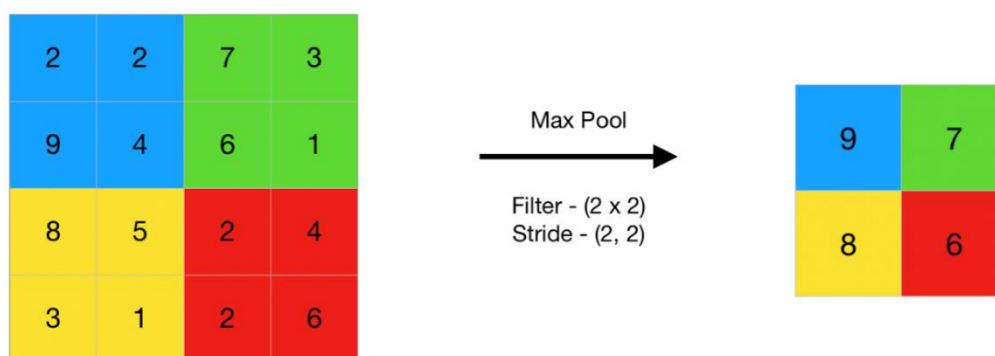


圖 2 最大池化層示意圖

2.2.2 Average pooling (平均池化) [6]

平均池化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。

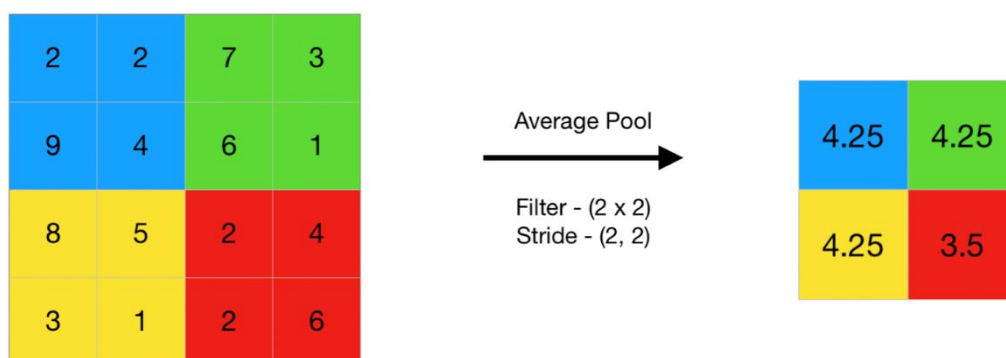


圖 3 平均池化層示意圖

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。

- 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

組員	內容	備註
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	

表 1 組員工作分配表

第二章 系統相關技術介紹

2.1 Python

Python 程式是一種易學、功能強大的程式語言。其具有高效能的高階資料結構，也有簡單但有效的方法去實現物件導向程式設計。Python 優化的語法和動態型別，結合其直譯特性，使它成為大數據分析、人工智慧等重要主流語言。而本組透過 Python 執行模型訓練、後端銜接。

2.2 Flask

2.2.1 Flask 簡介

Flask 是 Python 的輕量 Web 微框架[7]，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。

```
pip install flask
```

圖 4 Flask pip 安裝套件

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[8]。

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。
[9]

- ◆ 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。

2.2.4 Templates (動態文件)

專案文件夾中新增 templates 資料夾，可以將副檔名.html 的檔案放入，利用 render_template () 渲染模板可以將 html 檔進行渲染。[10]

2.2.5 Static (靜態文件)

若 flask 需要處理 static 靜態檔案 (CSS、Javascript、圖片)，需要在該專案資料夾中新增 static 資料夾，將欲加入的靜態檔案放入 static 資料夾中才可以使用。

2.2.6 url_for ()、redirect

透過 url_for 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。

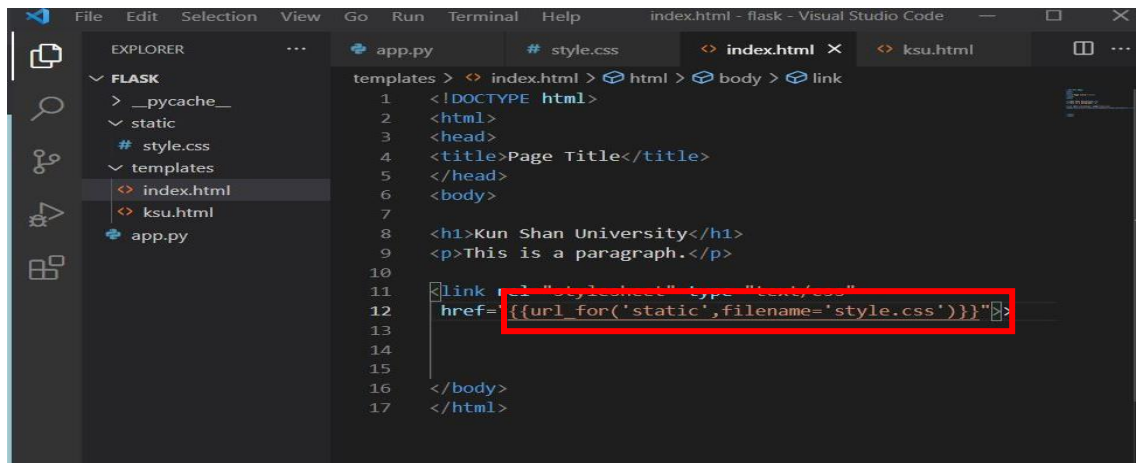
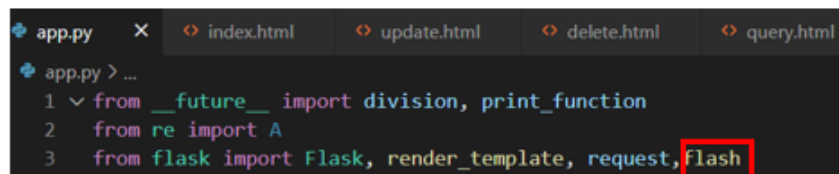


圖 5 url_for () 示意圖

2.2.7 flash message

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[11]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[11]，首先使用者需要先 import flash，flask 有用到 session 需要一組密碼進行 secret key 加密，透過迴圈來取得訊息，來判斷有無 message 訊息，例如：本專題的刪除功能，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。

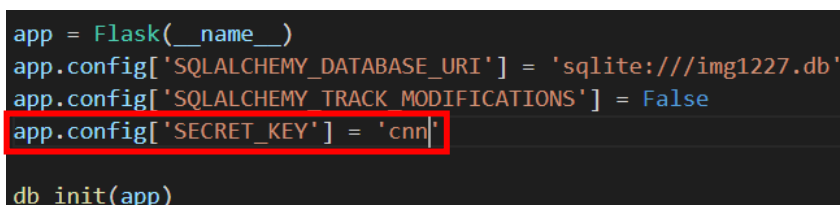


```

app.py x index.html update.html delete.html query.html
app.py > ...
1 from __future__ import division, print_function
2 from re import A
3 from flask import Flask, render_template, request, flash

```

圖 6 import Flash



```

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 7 設定 secret key 加密

```

{% 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 8 將迴圈程式新增到欲顯示的 html

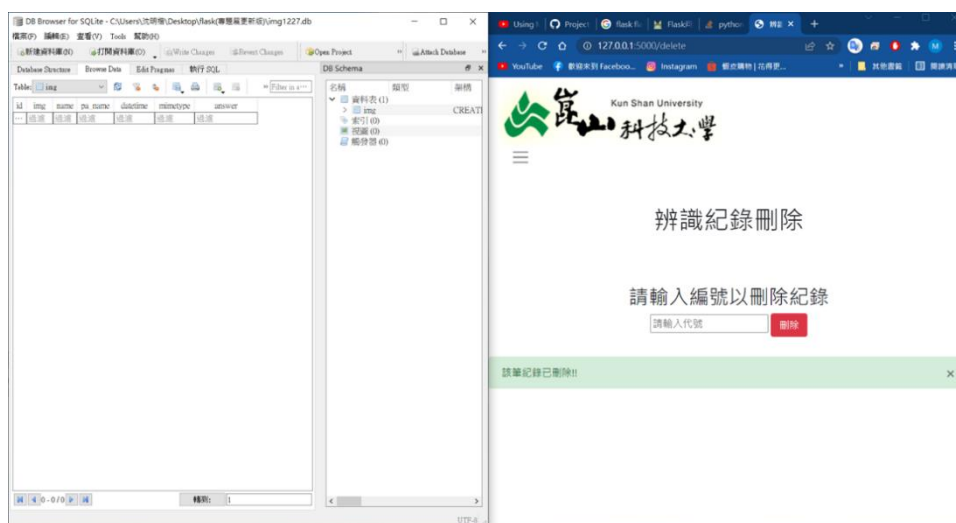


圖 9 刪除介面的 flash message

2.2.8 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。[12]

“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	使用者定義的導航條
content	使用者定義的頁面內容
scripts	文件底部的JS宣告

圖 10 繼承語法

例如：本專題的 home.html 需要給子模板[12] (upload.html) 繼承 (extends) 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。

圖 11 home.html 父模板

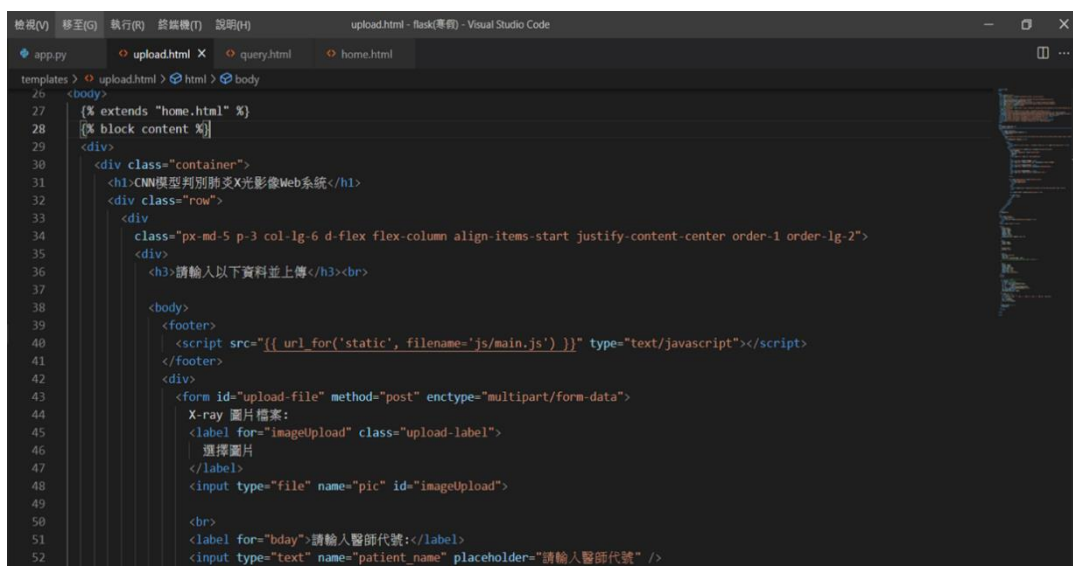


圖 12 upload.html 子模板



圖 13 upload.html 已繼承 home.html 的 navbar 導覽列

2.3 Flask-SQLAlchemy[13]

Flask 本身不支援 SQL 操作，SQLAlchemy 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。[13]

```
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 14 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)。

- ◆ 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'

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

表 2 資料庫示意表

2.4 SQLite 管理工具

DB Browser for SQLite：圖形化管理資料庫工具，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。

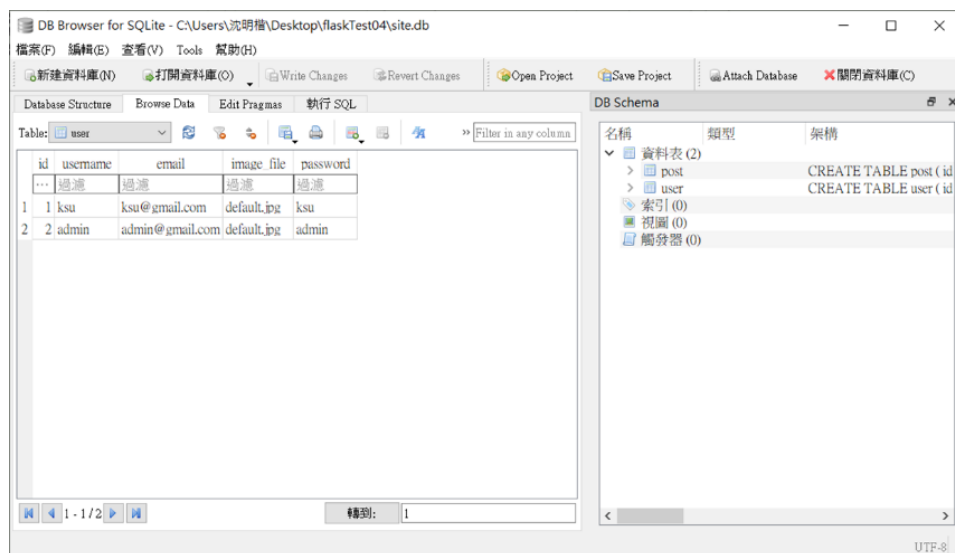


圖 15 DB Browser 軟體

BLOB(Binary Large Object)[14]用來儲存二進位檔案，像是影音、圖片或是大型文件，本專題利用 BLOB 格式儲存醫師從前端所上傳的 X 光片影像。[14]

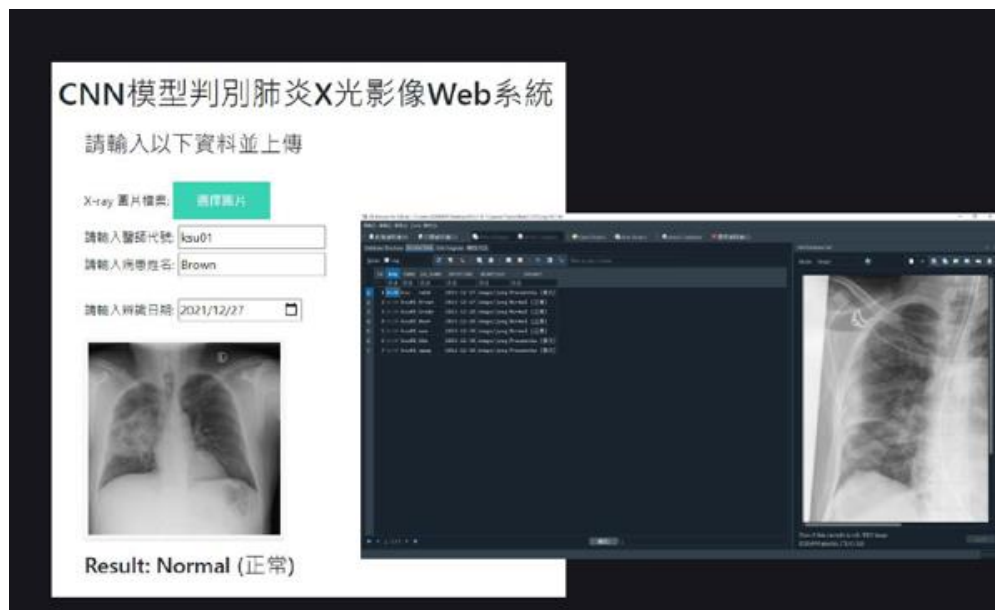


圖 16 醫師上傳 x 光至資料庫

第三章 系統流程與架構

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」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，以備特定醫師查看與後續動作。



圖 17 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	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 18 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。

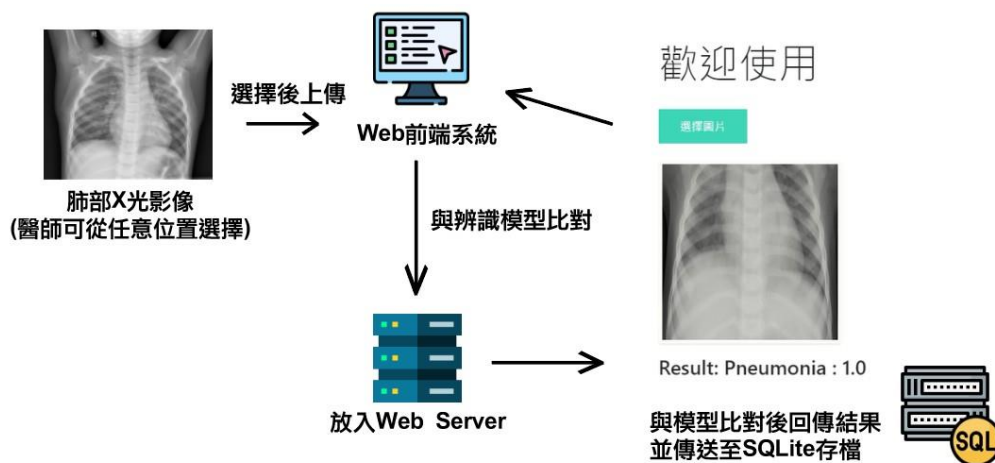


圖 19 系統架構圖

第四章 系統功能說明

4.1 醫學影像訓練子系統

4.1.1 醫學影像蒐集

資料集到開放式網站資源 Kaggle 下載到本地端。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：

TensorFlow Pneumonia Classification on X-rays

Python · [Chest X-Ray Images \(Pneumonia\)](#)

Notebook Data Logs Comments (53)

Data

xray_model.h5 (42.16 MB)

Download >

Unable to show preview



Previews for binary data are not supported

Input (1.24 GB)

Data Sources
 Chest X-Ray Images (Pn...
 chest_xray
 test
 train
 val

Output

xray_model.h5

```
> kaggle kernels output amyjang/tensorflow-pneumonia-classification-on-x-rays -p /pat_
```

圖 20 Kaggle Open Data[15]

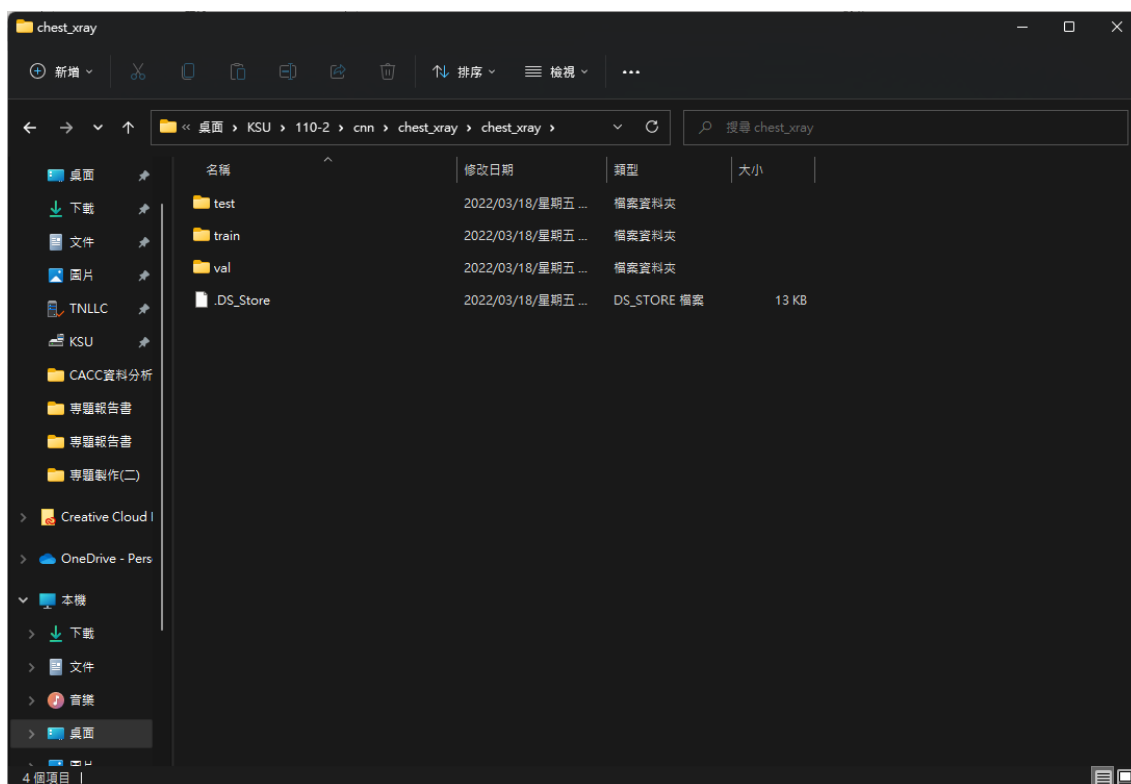


圖 21 將數據下載到本地電腦

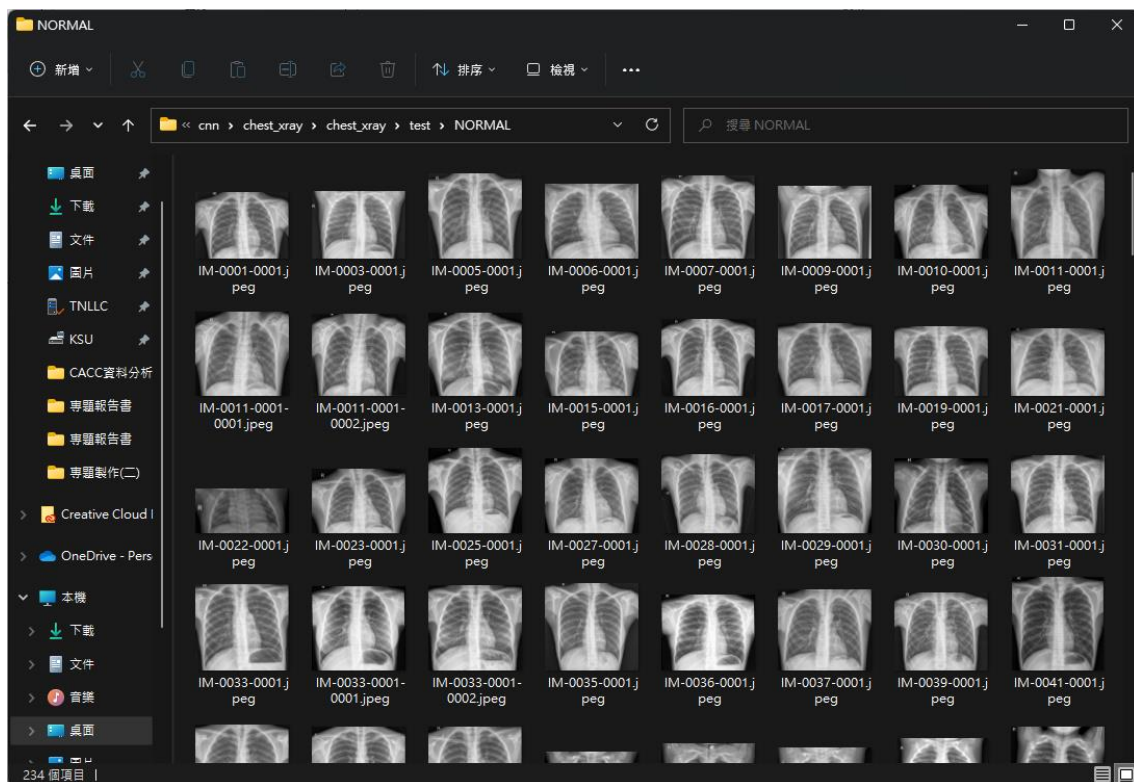


圖 22 測試集資料(正常肺部)

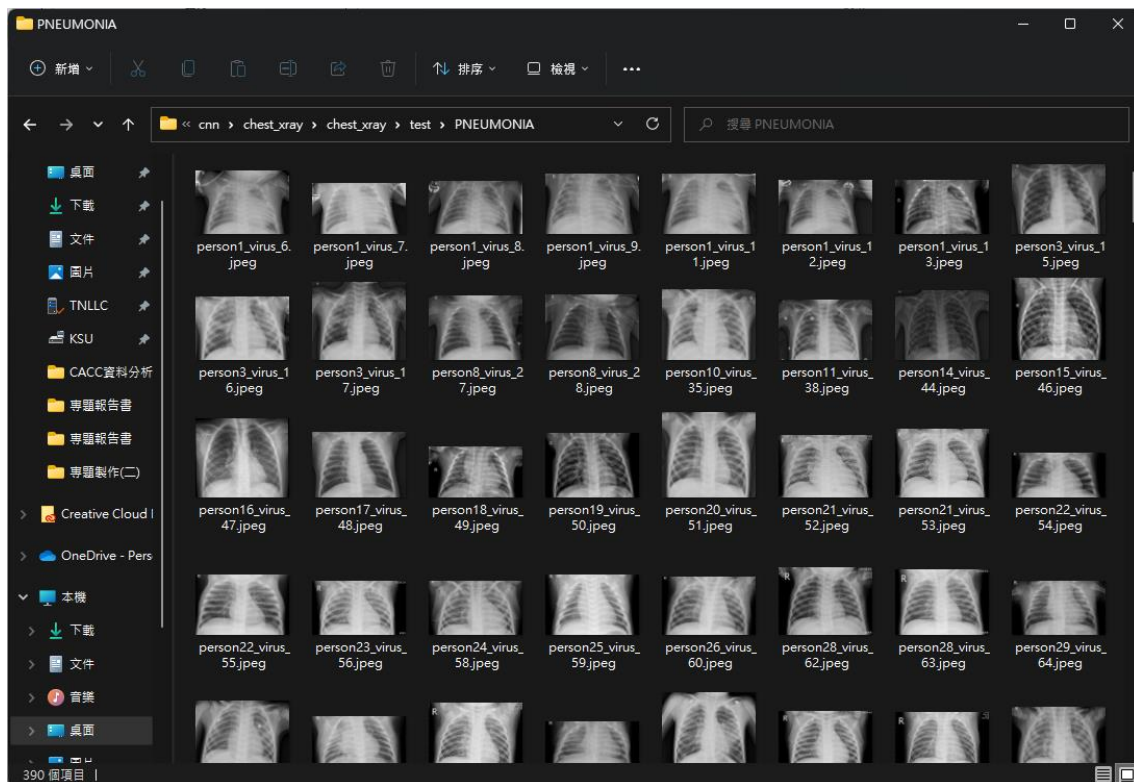


圖 23 測試集資料(罹患肺炎)

4.1.2 醫學影像模型訓練[16]

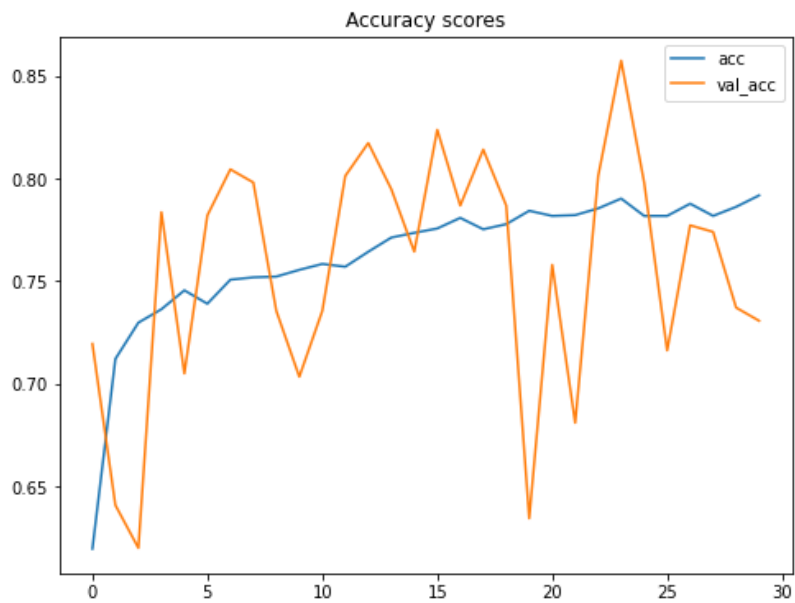


圖 24 影像訓練模型增益率

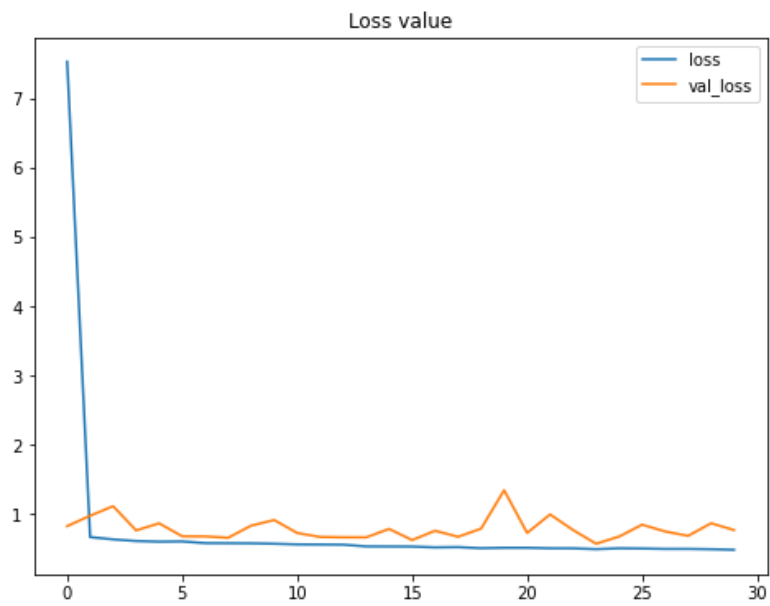


圖 25 影像訓練模型遺失率

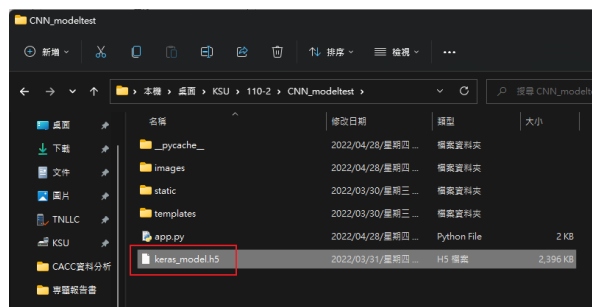


圖 26 模型訓練後產生之模型檔案(提供給伺服網頁使用)

4.2 使用者 Web 介面子系統

使用者(醫師)登入介面，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。

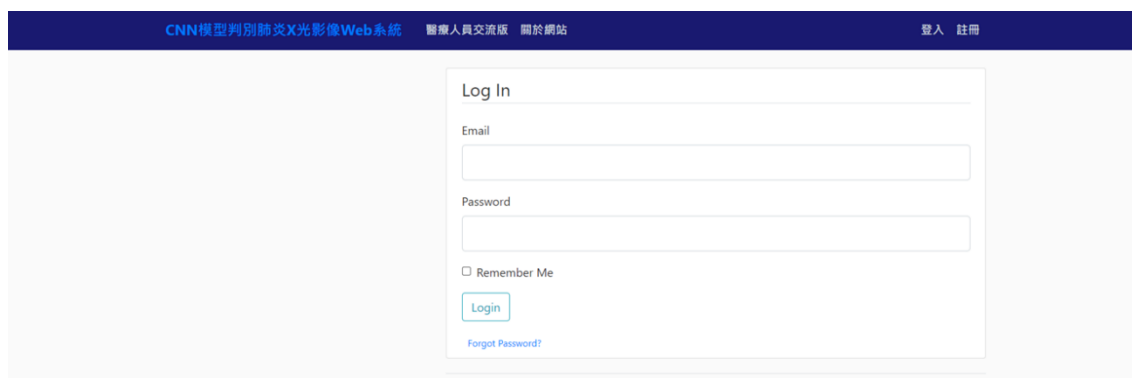


圖 27 使用者前端登入介面

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，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 28 登入後會依照帳號顯示醫生名稱

4.3 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。

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 29 辨識 X 光片介面

	id	img	name	pa name	datetime	mimetype	answer
	過濾	過濾	過濾	過濾	過濾	過濾	過濾
1	4	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)

資料表 (3)

- > img
- > post
- > user
- 索引 (0)
- 視圖 (0)
- 觸發器 (0)

圖 30 資料上傳至資料庫中

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。

例如身分為 ksu：

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

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 31 辨識紀錄查詢介面

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筆資料

	編號	圖片	醫師ID	病患姓名	日期	結果
編輯	5		ksu	007	20220414	Pneumonia (肺炎)

	編號	圖片	醫師ID	病患姓名	日期	結果
編輯	7		ksu	001	20220410	Normal (正常)

	編號	圖片	醫師ID	病患姓名	日期	結果
編輯	8		ksu	006	20220414	Normal (正常)

	編號	圖片	醫師ID	病患姓名	日期	結果
編輯	9		ksu	林小音	20220419	Normal (正常)


	編號	圖片	醫師ID	病患姓名	日期	結果
編輯	10		ksu	王阿宏	20220419	Normal (正常)

圖 32 辨識記錄列表

例如身為 ksu1：

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

歡迎登入，
ksu1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
- 刪除資料
- 修改資料

辨識紀錄查詢

點擊查詢按鈕以繼續(不能修改)

ksu1

更多...	編號	圖片	醫師ID	病患姓名	日期	結果
	4		ksu1	041	2022-04-10	Normal (正常)

圖 33 ksu1 身分登入頁面

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：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 34 編輯頁面畫面，並點選 Update 按鈕，但是只有病患姓名與上傳日期能更改



圖 35 修改病患姓名



圖 36 Flash message 顯示已更新，資料庫也更新成功

- Delete 刪除按鈕：使用者想要刪除某一筆辨識的歷史紀錄，可點選編號左邊編輯的按鈕，即可進入編輯介面，點選 Delete 刪除按鈕，系統會跳出警告進一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 37 刪除前之資料庫

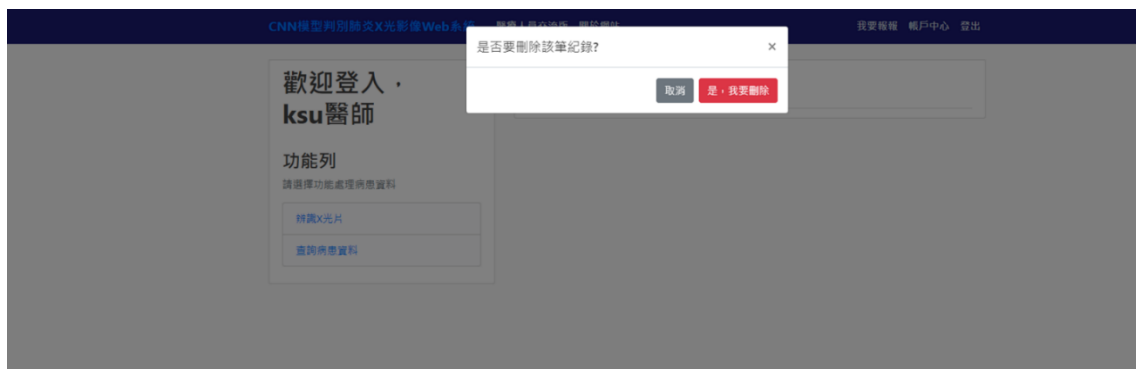


圖 38 系統 alert 警告是否刪除該筆資料



圖 39 Flash message 顯示該筆紀錄已刪除

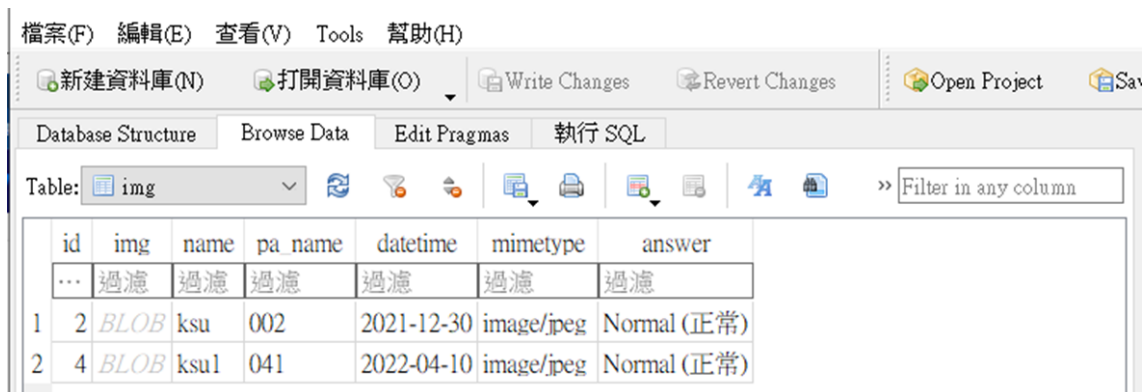


圖 40 該筆資料已被刪除



圖 41 重新進入查詢也發現該筆紀錄已刪除

第五章 結論

第六章 參考文獻

1. Convolutional Neural Networks (CNNs / ConvNets) ,
<https://cs231n.github.io/convolutional-networks/>
2. 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
3. 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>
4. 卷積神經網路 (Convolutional Neural , CNN)，李宏毅，
<https://hackmd.io/@allen108108/rkn-oVGA4>
5. A Gentle Introduction to Pooling Layers for Convolutional Neural Networks，
Jason Brownlee，<https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>，2019/04/22
6. Average Pooling，<https://paperswithcode.com/method/average-pooling>
7. Flask，維基百科，<https://zh.m.wikipedia.org/zh-tw/Flask>
8. Flask-Routing，<https://flask.palletsprojects.com/en/2.1.x/quickstart/#routing>
9. Flask HTTP methods, handle GET & POST requests，Python Tutorial，
<https://pythonbasics.org/flask-http-methods/>
10. Flask-Template，<https://flask.palletsprojects.com/en/2.1.x/quickstart/#rendering-templates>
11. Flask 實作_基礎_10_flash message，
<https://hackmd.io/@shaoeChen/BkkbvYIff?type=view>
12. Child Template，
<https://flask.palletsprojects.com/en/1.1.x/patterns/templateinheritance/>
13. Flask-SQLAlchemy，<https://flask-sqlalchemy.palletsprojects.com/en/2.x/>
14. Binary Large Objects，
https://docs.oracle.com/cd/E17276_01/html/api_reference/C/blob.html
15. TensorFlow Pneumonia Classification on X-rays，
<https://www.kaggle.com/code/amyjang/tensorflow-pneumonia-classification-on-x-rays/data>
16. 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