**中 國 文 化 大 學**

**資 訊 工 程 學 系**

**資 訊 系 統 專 題**

**校園停車場管理系統**

**學 生：卓 品 佑**

**陳 思 釩**

**指導教授：李 志 仁**

**中 華 民 國 108年 5 月**

**校園停車場管理系統**

**專題學生：卓品佑、陳思釩**

**指導教授：李志仁**

**中國文化大學 資訊工程學系**

**中文摘要**

很多學校的停車場管理運作模式是申請卡式停車證並憑卡入場制，而我們想要使流程變得更為簡便，因此本專題我們會製作一個校園停車場管理系統，用來管理已申請學校停車資格且繳費完畢的學生車輛，先從攝影機擷取車牌圖像並辨識出車牌號碼，再與資料庫中的學生車牌資料做比對，如果非申請停車資格或已申請但尚未繳費的車輛進入停車場，系統將會提出警告，我們把車牌辨識和管理學生資料的功能分開，車牌辨識功能使用 python的opencv模組來製作，學生車牌資料存放在mysql，並使用php編寫的網頁來管理學生資料。

**關鍵詞：**影像處理、資料庫系統

指導教授\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(簽名)

**目 錄**

[中文摘要 I](#_Toc10019752)

[目 錄 II](#_Toc10019753)

[圖 目 錄 IV](#_Toc10019754)

[表 目 錄 V](#_Toc10019755)

[第1章 研究動機與目的 1](#_Toc10019756)

[1.1 研究動機 1](#_Toc10019757)

[1.2 研究目的 1](#_Toc10019758)

[第2章 文獻討論 2](#_Toc10019759)

[2.1 影像處理的相關研究 2](#_Toc10019760)

[2.1.1 彩色模式 2](#_Toc10019761)

[2.1.2 灰階 2](#_Toc10019762)

[2.1.3 二值化 3](#_Toc10019763)

[2.1.4 邊緣偵測 3](#_Toc10019764)

[2.1.5 樣板比對 4](#_Toc10019765)

[2.2 所使用工具的介紹與比較 5](#_Toc10019766)

[2.2.1 Open CV介紹 5](#_Toc10019767)

[2.2.2 Python介紹 5](#_Toc10019768)

[2.2.3 工具的比較 5](#_Toc10019769)

[第3章 資料庫系統 6](#_Toc10019770)

[3.1 資料庫 6](#_Toc10019771)

[3.2 網頁實作 8](#_Toc10019772)

[3.3 車牌辨識系統介面 9](#_Toc10019773)

[第4章 車牌辨識 14](#_Toc10019774)

[4.1 影像處理 15](#_Toc10019775)

[4.1.1 灰階值 16](#_Toc10019776)

[4.1.2 高斯模糊 17](#_Toc10019777)

[4.1.3 中值濾波 18](#_Toc10019778)

[4.1.4 Sobel邊緣檢測 19](#_Toc10019779)

[4.1.5 二值化 20](#_Toc10019780)

[4.1.6 侵蝕與膨脹 21](#_Toc10019781)

[4.2車牌偵測 24](#_Toc10019782)

[4.3字元分割 25](#_Toc10019783)

[4.4字元辨識 25](#_Toc10019784)

[4.5本專題使用工具 25](#_Toc10019785)

[第5章 人力配置 27](#_Toc10019786)

[第6章 預期成果 28](#_Toc10019787)

[第7章 執行進度 29](#_Toc10019788)

[參考文獻 30](#_Toc10019789)

**表　目　錄**

[表4.1 工具列表 26](#_Toc10112997)

[表5.1參與本項研究之人員及工作內容 27](#_Toc10112998)

[表7.1 專題進度表 29](#_Toc10112999)

**圖 目 錄**

[圖2.1 影像處理步驟圖 3](#_Toc10113110)

[圖2.2 兩個同質但不同色的區域 4](#_Toc10113111)

[圖2.3 樣板圖示 5](#_Toc10113112)

[圖3.1 資料庫架構圖 7](#_Toc10113113)

[圖3.2 phpmyadmin截圖 8](#_Toc10113114)

[圖3.3 學生資料管理網頁架構圖 9](#_Toc10113115)

[圖3.4 車牌辨識系統介面截圖 10](#_Toc10113116)

[圖3.5 車牌辨識系統分頁介面設計圖 11](#_Toc10113117)

[圖3.6 學生資料查詢分頁介面設計圖 12](#_Toc10113118)

[圖3.7 車牌辨識系統設計圖 13](#_Toc10113119)

[圖4.1 車牌辨識步驟圖 14](#_Toc10113120)

[圖4.2 影像處理步驟圖 15](#_Toc10113121)

[圖4.3 車牌影像原圖 16](#_Toc10113122)

[圖4.4 轉換成灰階圖像後 17](#_Toc10113123)

[圖4.5 高斯模糊後的圖像 18](#_Toc10113124)

[圖4.6 中值濾波後得到的圖像 19](#_Toc10113125)

[圖4.7 Sobel運算後的圖像 20](#_Toc10113126)

[圖4.8 二值化後的圖像 21](#_Toc10113127)

[圖4.9 膨脹一次後的圖像 22](#_Toc10113128)

[圖4.10 侵蝕後的圖像 23](#_Toc10113129)

[圖4.11 二度膨脹後的圖像 23](#_Toc10113130)

[圖4.12 進行車牌偵測後得到的圖像 24](#_Toc10113131)

**第1章 研究動機與目的**

**1.1 研究動機**

根據交通部統計的資料顯示，108年2月的機車數量為1384萬9120，汽車數量為804萬2165，隨著每年的人均車輛數比例增長，大量汽機車產生的問題也就越來越多，例如違停、肇事逃逸、車輛協助犯罪等，使車輛管理這份工作顯得十分重要

車牌辨識廣泛運用在停車場管理、交通違規、失竊車輛追蹤、肇事車輛逃逸、警方偵查犯人車輛等，在各種環境干擾因素下擷取車牌影像並分析出車牌號碼，本專題我們會利用影像處理技術來製作車牌辨識系統，進而達到控管車輛的目的。

汽車和機車已成為主要的交通工具，但是在停車場，傳統的人工管理費時又費力，不符合經濟成本，本專題研究對車輛的自動化管理，節約時間與人力，提高了工作效率。

**1.2 研究目的**

為了幫助學校更好的管理停車場，本專題利用資料庫記錄學生的繳費訊息，將進入的車牌辨識後與資料庫對比，判斷是否具有停車資格，這樣一來，保障了繳費學生的權益，讓同學有一個更加安全，更加方便的停車環境，同時也大大減輕了學校管理人員的負荷。

**第2章 文獻討論**

基於第1章所述之研究動機與目的，我們對目前現有車牌辨識系統以及會使用到的技術和工具進行進一步的探討。

**2.1 影像處理的相關研究**

影像處理[1]是指對圖像進行分析、加工、和處理，使其滿足視覺、心理或其他要求的技術。大致可以分為七個步驟：

**2.1.1 彩色模式**

在影像模式中，RGB、YIQ、HSV、YUV、YcbCr這些是較為常見的彩色模式，各種彩色模式有不同的特性，例如電腦是以 RGB 做為顯色的依據，因此對於每一種彩色模型都必須換到 RGB 模式才可令它顯色,另外 HSV 彩色模式可以減少因為亮度變化造成的影響。RGB就是red、green、blue三種顏色的縮寫。也就是一般所稱的三原色,這三原色可以製作各種不同的顏色，只要依據其 RGB 大小的組合即可。[2]

**2.1.2 灰階**

用於顯示的灰階圖像通常用每個採樣像素8 bits的非線性尺度來保存，這樣可以有256種灰階（8bits就是2的8次方=256），這種精度剛剛能夠避免可見的條帶失真，並且非常易於編程。[3]如圖2.1所示。從彩色影像轉換成灰階影像的過程稱為灰階化，通過形成的R、G和B分量的加權總和： 0.2989\*R + 0.5870\*G + 0.1140\* B。[4]

圖2.1 影像處理步驟圖

**2.1.3 二值化**

二值化可以把灰度圖像轉換成二值圖像，把大於某個臨界灰度值的像素灰度設為灰度極大值，把小於這個值的像素灰度設為灰度極小值，從而實現二值化[5]。根據閾值選取的不同，二值化的算法分為固定閾值和自適應閾值，簡單通俗的說，二值化的作用就是將圖像分成黑和白，更加有利於做圖像處理判別。

**2.1.4 邊緣偵測**

邊緣偵測的目的是標識數字圖像中亮度變化明顯的點。圖像屬性中的顯著變化通常反映了屬性的重要事件和變化。利用相鄰像素的差距找出物體的邊緣,因此像素差距越明顯就能更清楚的表達出邊緣。找出正確、明顯的邊緣就能更有效率的辨識車牌位置。車牌會因為拍攝角度不同,而產生不一樣的形狀 ,也會因為天氣、亮度造成拍攝的結果而有所不一樣。[6]

圖2.2 左半部是黑色,而右半部是白色。在介於兩個同質(Homogeneous)但不同色的邊緣處(Boundary)會出現大的灰階圖變化。然而,在個別的區域裡內,鄰近的灰階值很相近。這灰階的突然變化(Abrupt Change)即是測邊的主要觀念之一。[2]



圖2.2 兩個同質但不同色的區域

**2.1.5 樣板比對**

形狀檢測辨識最直接的方法，就是把所有標準圖形事先儲存在電腦裡。當電腦遇到一個待測圖形時，就把該圖形與所有事先儲存的圖形庫資料一一叫出來比對，比對出來最接近該待測圖形的標準圖形即可達到形狀檢測辨識的目的。

而簡單的比對方法,就是把兩個圖形放置在同一個位置上,然後令兩個影像相減，或是計算兩個影像之間重疊的面積，通常這樣做也會耗去很多時間。這樣比對的方法常常需要精確的定位，否則兩個影像也許只是位置錯開了，但內容是相同的，相減後許多點素仍會殘留灰度值，如此會把同一個影像的兩種不同形式判定為兩個不同的影像。

因此在形狀檢測辨識以前應該對標準圖形作一些結構的分析，然後根據結構的特徵來從事形狀檢測辨識與比對，而樣板(template)即可視為具有影像結構特徵之子影像，利用樣板來從事形狀比對的工作，有助於簡化整個形狀檢測辨識的流程。[6] 如圖2.3所示。

圖2.3 樣板圖示

**2.2 所使用工具的介紹與比較**

**2.2.1 Open CV介紹**

OpenCV 的全稱是 Open Source Computer Vision Library，是一個跨平台的 計算機視覺庫。OpenCV是由英特爾公司發起並參與開發，以 BSD 許可證 授權發行，可以在商業和研究領域中免費使用。OpenCV 可用於開發實時 的圖像處理、計算機視覺以及模式識別程序。該程序庫也可以使用英特爾 公司的 IPP 進行加速處理[7]。

**2.2.2 Python介紹**

Python是一種廣泛使用的直譯式、進階編程、通用型程式語言，由吉多·范羅蘇姆創造，第一版釋出於1991年。可以視之為一種改良（加入一些其他程式語言的優點，如物件導向）的LISP。Python的設計哲學強調程式碼的可讀性和簡潔的語法（尤其是使用空格縮排劃分程式碼塊，而非使用大括號或者關鍵詞）。相比於C++或Java，Python讓開發者能夠用更少的代碼表達想法。不管是小型還是大型程式，該語言都試圖讓程式的結構清晰明了[8]。

**2.2.3 工具的比較**

本專題影像處理部分我們使用的是Python+Open CV，它的優點如下：語法簡單、易於使用、**視覺化及除錯**。

[6]中使用了Visual Studio+Open CV，而我們不使用Visual Studio+Open CV是因爲，一個計算機視覺引擎時常需要大量的機器學習程式，相比於Python+Open CV，Visual Studio+Open CV，只含有一個小型的機器學習演算法子集。在任何一個 C 的開發環境中，我們都難以進行除錯和視覺化。尤其是對於那些在忙亂中產生的新演算法來說，我們更能體會到這點。

[4]中使用的是Matlab，雖然Matlab運算功能强大，工具箱有影像處理函式，操作面板簡易，但是相對於Python+Open CV，Matlab執行速度較慢，成本也比較大。

**第3章 資料庫系統**

我們利用mysql把學生資料放在資料庫STUDENT的資料表information中，並實作php網頁來操作這些資料，車牌辨識功能則是使用python的opencv模組來實現，並使用pymysql模組來連接mysql，把辨識出來的車牌號碼與繳費名單做對比。

**3.1 資料庫**

我們建立一個資料庫STUDENT，在STUDENT內新增資料表imformation，imformation內新增6個欄位，包括姓名(name)、學號(sid)、車牌(license\_plate)、年級(grade)、系級(department)、繳費情況(pay)，圖3.1是我們建立的資料庫ER圖。

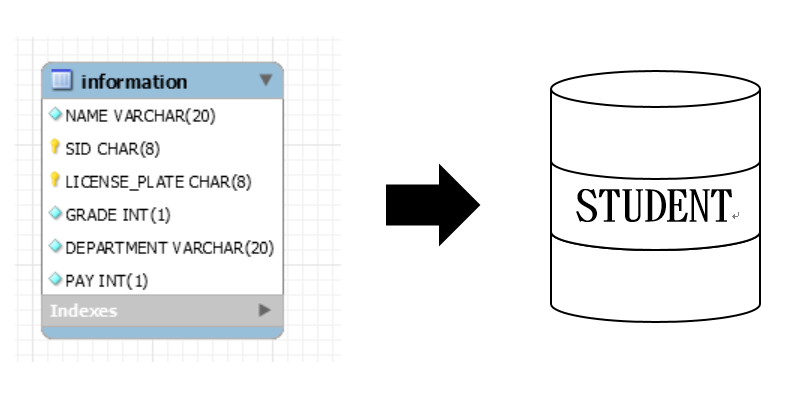


圖3.1 資料庫架構圖

學號和車牌欄位設定為主鍵，主鍵是用來辨識每筆資料的依據，所以主鍵必須是唯一性，而學號和車牌符合這設定，故把這兩個欄位設定為主鍵。

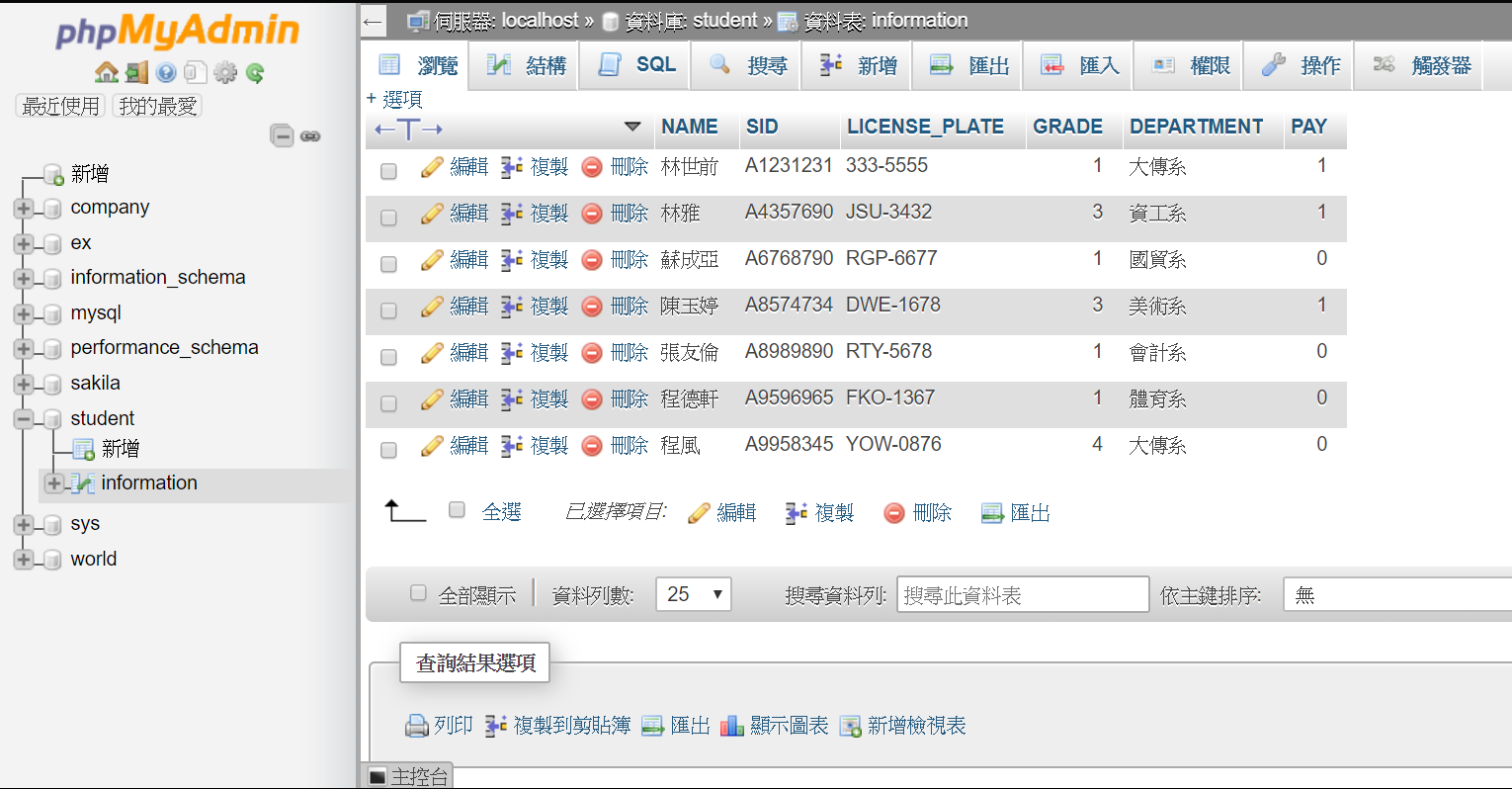
之後使用phpmyadmin來讓php編寫的網頁可以連接到mysql，圖3.2為phpmyadmin連接mysql的截圖。

圖3.2 phpmyadmin截圖

左側是資料庫選單，操作功能在上列，中間的部分是information資料表裡面的內容，phpmyadmin可以讓使用者用網頁來操作mysql，十分方便。

**3.2** **網頁實作**

我們的php文件配置，主介面連接五個選項，分別是新增、刪除、修改、查詢和查看學生名單，圖3.3是網頁架構圖。

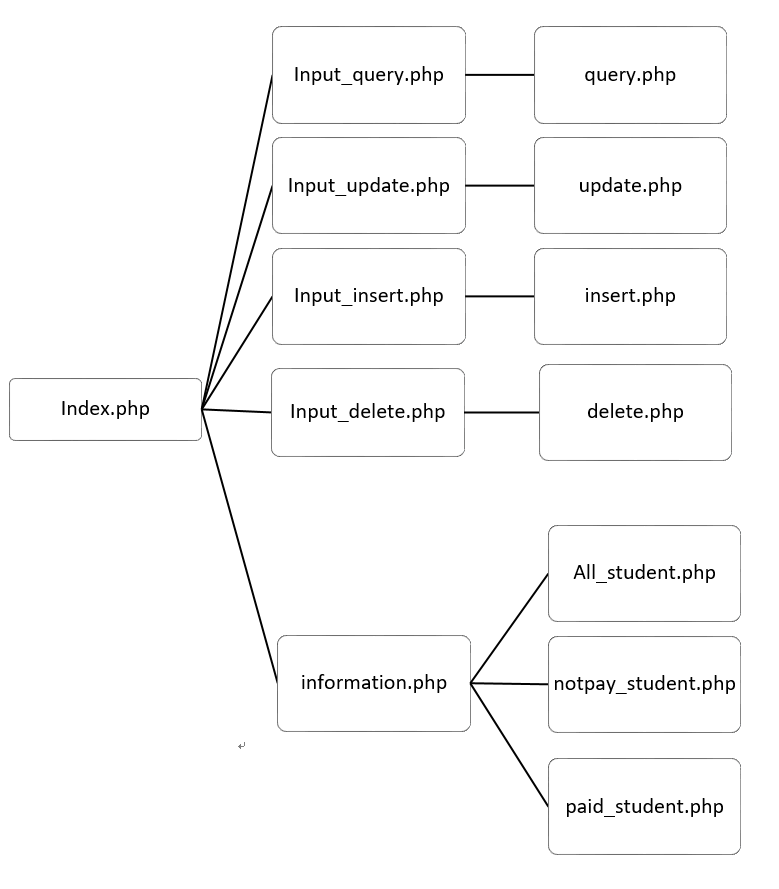


圖3.3 學生資料管理網頁架構圖

我們把新增、刪除、修改、查詢四個功能各自再分成兩個檔案，分別是輸入介面和本身的功能實作，而學生名單再細分成全體學生、已繳費學生和未繳費學生三個選項。

**3.3 車牌辨識系統介面**

使用python內建程式庫的tkinker模組和tkinter中的ttk組件製作系統介面，圖3.4為車牌辨識系統的介面，並設有兩個分頁， 分別設置了車牌辨識功能和查詢功能。

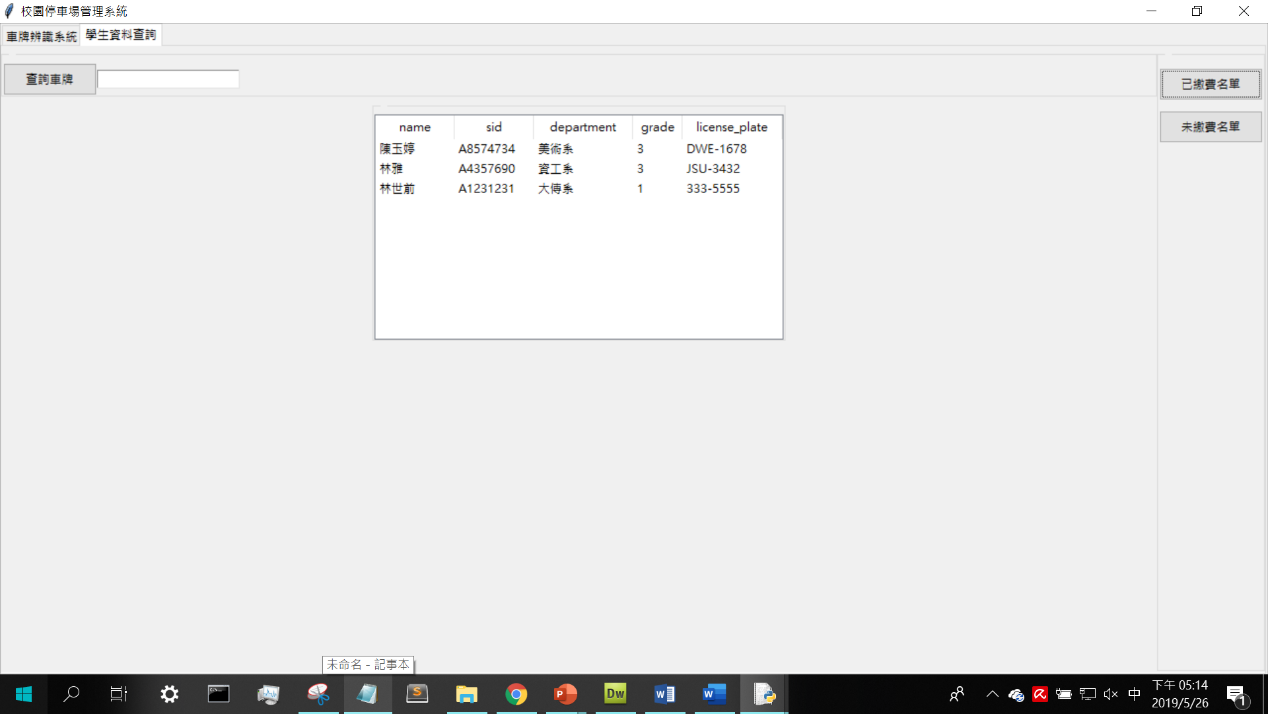


圖3.4 車牌辨識系統介面截圖

這裡展示的是學生資料查詢的分頁，點已繳費按鈕後顯示已繳費名單，而只要在上面的輸入框輸入車牌再按下查詢車牌按鈕，輸入框的右側便會依照該車輛是否申請停車格或是否已繳費來出現應對的文字敘述。

圖3.5是車牌辨識系統分頁的界面設計，攝影機開關按鈕和辨識結果放在right labelframe，left label顯示攝影機傳入的影像。

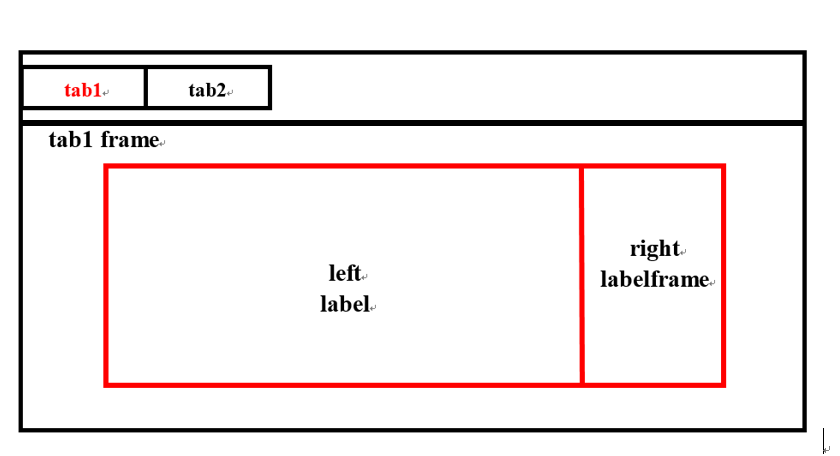


圖3.5 車牌辨識系統分頁介面設計圖

而查詢功能放在學生資料查詢這個tab，查詢功能有輸入車牌來查詢繳費情況和查看已繳費與未繳費的學生名單，如圖3.6已繳費與未繳費的學生名單按鈕放在right labelframe，結果顯示在left labelframe2，查詢功能放在left labelframe1。

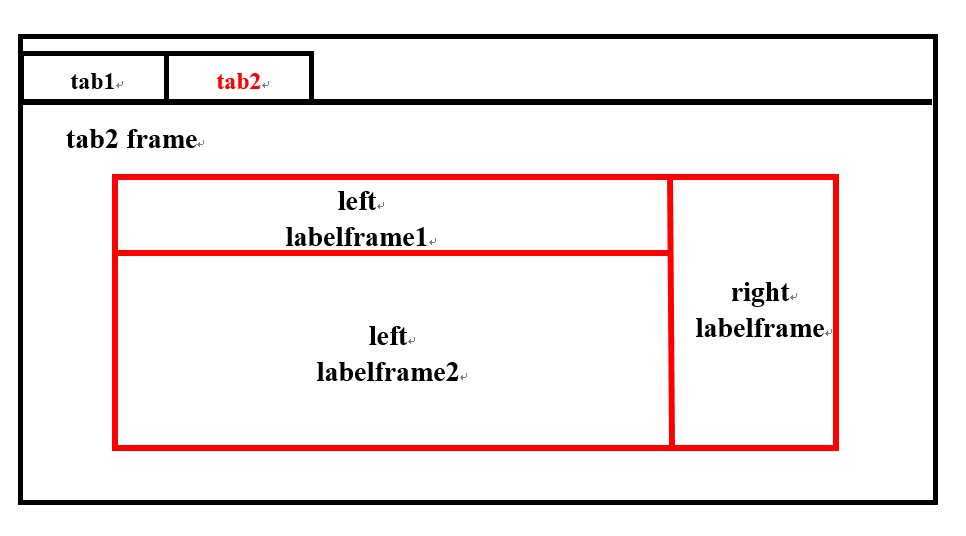


圖3.6 學生資料查詢分頁介面設計圖

完整的系統設計圖如圖3.7所示，車牌影像輸入到系統後開始車牌辨識，辨識結果和mysql的資料做對比，未繳費或未申請停車資格的車輛近入停車廠時系統將提出警告。

車牌影像

車牌號碼

車牌辨識系統

車號號碼

判斷車輛是否有申請停車格

判斷車輛是否有申請停車格

車輛若無申請停車格，

系統提出警告

車輛若未繳費，

系統提出警告

車輛若已繳費，

系統提示可進場

圖3.7 車牌辨識系統設計圖

**第4章 車牌辨識**

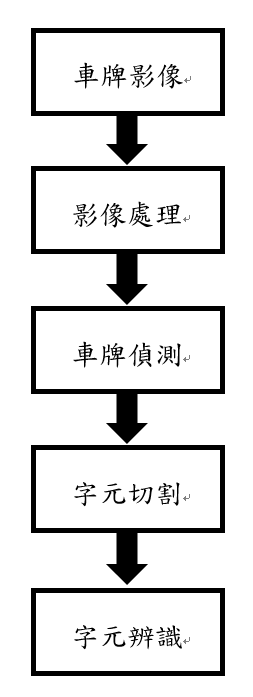
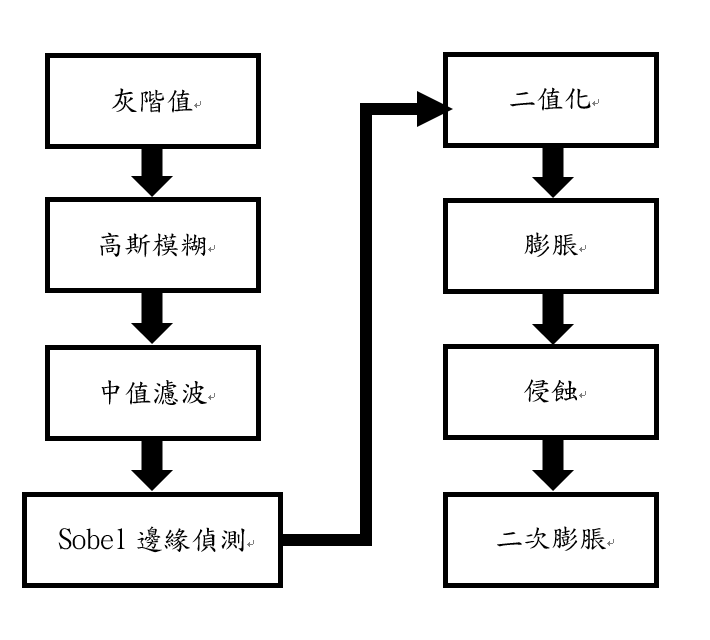
車牌辨識功能會細分成幾個步驟來實現，我們的車牌辨識步驟如圖4.1。

圖4.1 車牌辨識步驟圖

攝影機擷取車牌影像並匯入車牌系統，影像經過一些處理後去除背景只留車牌圖像，再來分割車牌中的字元，最後做字元辨識。

**4.1 影像處理**

我們得設計出能讓車牌部分和背景有鮮明對比的影像處理演算法才能利用輪廓擷取出車牌圖像，圖4.2是影像處從圖4.2來看，演算法由opencv的函式庫所構成。

圖4.2 影像處理步驟圖

**4.1.1 灰階值**

使用cvtColor函數使彩色影像的圖4.3轉變成灰階影像，cvtColor函數格式為cv2.cvtColor(input\_image, flag)，flag指的是轉換類型，彩色轉灰階圖像的轉換類型為COLOR\_BGR2GRAY。

圖4.3 車牌影像原圖

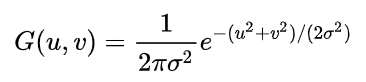
運算公式為Gray = R\*0.299 + G\*0.587 + B\*0.114，彩色圖像的紅、綠、藍色在灰階圖片中以亮度做區分，圖4.4為灰階化後的圖像。



圖4.4 轉換成灰階圖像後

**4.1.2 高斯模糊**

使用GaussianBlur函數來去除影像中的雜訊GaussianBlur函數的運作原理是每一个像素都取周边像素的平均值[9]，函數格式為cv2.GaussianBlur (input\_image, ksize, sigma) ，ksize指的是核心的大小，sigma指的是標準差，高斯矩陣[10]或標準差越大圖像會越模糊，圖4.5為經過高斯模糊處理後的圖像，圖像經由高斯模糊處理後變得更加模糊。

越靠近a的點，影響的權重越多

\* \* \*

\* a \*

\* \* \*

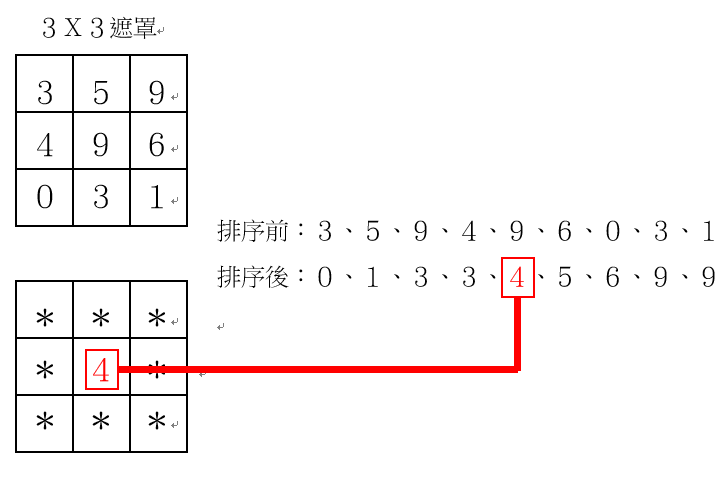


圖4.5 高斯模糊後的圖像

**4.1.3 中值濾波**

使用medianBlur來去除影像中的雜訊，medianBlur 函數格式為cv2.medianBlur(input\_image, ksize)，中值濾波[11]的運作原理是某一點的輸出值用這點的某個大小的鄰域內的所有值的統計中值來代替。鄰域越大輸出的結果就越平滑，此外ksize(核心)必須為奇數才有唯一的中間值，圖4.6為經過中值濾波處理後的圖像。

中值濾波的計算方式是先把遮罩內的像素值做排列，再取中間的像素值，中值濾波如下：



****

圖4.6 中值濾波後得到的圖像

**4.1.4 Sobel邊緣檢測**

使用Sobel運算子垂直方向檢測車牌邊緣，Sobel函數的格式為cv2.Sobel(input\_image, ddepth, dx, dy, ksize)，ddepth[12]指的是輸出圖的深度，dx指的是x方向的微分階數，dy則是y方向的微分階數，本專題使用x方向做垂直檢測，圖4.7為經過Sobel運算後的圖像，車牌號碼字元的兩側變得十分明顯。

圖4.7 Sobel運算後的圖像

**4.1.5 二值化**

使用threshold函數做二值化， threshold函數格式為cv2.threshold

(gray\_image, thresh, maxval, type)，二值化之前一定要先把圖形轉換成灰階圖像，thresh指的是閾值，maxval指的是二值化結果的最大值，type指的是二值化操作類型，本專題使用的是cv2.THRESH\_BINARY，超過閾值的像素設為最大值maxval，小於閾值的設為0 [12]，圖4.8為經過二值化處理後的圖像，此時圖像中原本的雜訊已經去除了。

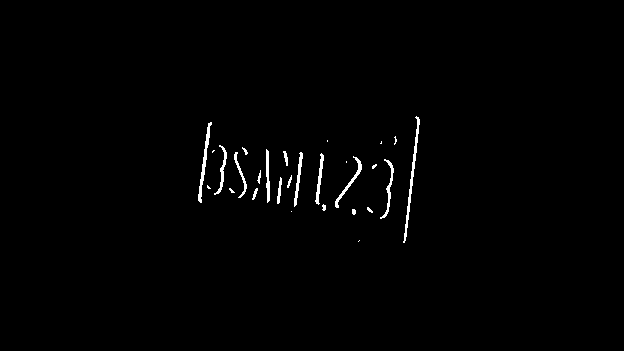


圖4.8 二值化後的圖像

**4.1.6 侵蝕與膨脹**

我們先使用dilation函數進行一次膨脹，dilation函數格式為cv2.

dilate(img,kernel,iterations)，kernel指的是結構元素，本專題使用的kernel為9x1，kernel越大膨脹效果越明顯，iterations指的是執行次數，默認為1次，圖4.9為一次膨脹後的圖像，膨脹處理後白色部分變大了。



圖4.9 膨脹一次後的圖像

接下來使用erode函數進行侵蝕，圖4.10為侵蝕後的圖像，erode函數的格式和dilation一樣，作用是使物體縮水，縮水的程度由kernel的大小來決定，侵蝕後再次執行膨脹函數，二次膨脹的圖像如圖4.11所示，這時白色部分填滿了車牌區域。



圖4.10 侵蝕後的圖像



圖4.11 二度膨脹後的圖像

白色區域即為我們所求的車牌位置，圖像前置處理的部分到此為止，接下來要進行車牌偵測的步驟。

**4.2車牌偵測**

先使用findContours函數尋找輪廓，findContours的函數格式為contours,hierarchy=cv2.findContours(thresh\_image, mode, method)，findContours只接收二值化圖像，mode指的是輪廓獲取模式，本專題用的是cv2.RETR\_TREE，cv2.RETR\_TREE會建立一個等級樹結構的輪廓。

而method指的是輪廓的近似方法，本專題使用

cv2.CHAIN\_APPROX\_SIMPLE，cv2.CHAIN\_APPROX\_SIMPLE壓縮水平方向、垂直方向、對角線方向的元素，只保留該方向的終點坐標，例如一個矩形輪廓只需4個點來保存輪廓信息[13]。

並使用contourArea函數計算輪廓面積，把面積小的輪廓都篩選掉，再來使用minAreaRect函數找出最小的矩形，確定比例後找出車牌區域並切割，圖4.12為進行切割車牌後的圖像。



圖4.12 進行車牌偵測後得到的圖像

可以見到其實也把車牌框和多餘的文字一併截進去了，此外也需要做角度調整，所以我們必須克服這個問題。

**4.3字元分割**

我們目前的想法是利用字元的間隔來分割字元，灰階處理後再二值化，讓車牌背景變成白色而字元變成黑色，字元之間的空隙黑色像素的數量會趨近於0，因此可以利用這點切割，把每一列的黑色像素加總，只要數量極少就有可能是字元的間隔。

**4.4字元辨識**

光學字元辨識（英語：Optical Character Recognition，OCR）是指對文字資料的圖像檔案進行分析辨識處理，取得文字及版面資訊的過程[14]，而我們將會使用Tesseract的方式做字元辨識的功能，Tesseract是一個OCR庫，可以通過訓練識別出任何字型。

**4.5本專題使用工具**

opencv支持python和C++語言，由於python的優勢是具有豐富的函式庫，故本專題使用python語言做開發工具，而dreamweaver的好處是使用者可以利用裡面自帶的工具來製作網頁，因此不太需要花時間在html語言的編寫上，表4.1即為本專題使用的工具列表。

表4.1 工具列表

|  |  |
| --- | --- |
| **工具** | **功能** |
| python-opencv | 使用opencv的python模組開發實時的圖像處理、電腦視覺以及圖形辨識程式 |
| MySQL | 建立資料庫儲存資料 |
| phpmyadmin | 利用PHP語言操作MySQL資料庫 |
| Dreamweaver | 製作網站 |

**第5章 人力配置**

參與本研究之工作人員及預定之工作內容如表5.1所示。卓品佑組長負責主要程式的編寫，陳思釩同學負責協助工作和整理資料，論文由兩人合作完成。

表5.1參與本項研究之人員及工作內容

|  |  |
| --- | --- |
| **姓名** | **工作內容** |
| 卓品佑 | 架設資料庫、php網頁、車牌系統程式、論文 |
| 陳思釩 | 車牌系統程式、整理資料、論文 |

**第6章 預期成果**

車牌辨識很常受到外在因素的干擾，例如光線、角度、車輛行駛速度，這些需要研發一套可以解決這些問題的演算法，閱覽過許多的車牌辨識系統相關論文，他們也都在為想辦法解決這些問題，由於我們希望能製作出辨識度較為精準的車牌辨識系統，因此需要在演算法的部分好好的鑽研，預期成果是車牌辨識系統能準確辨識出學生車輛的車牌號碼，網頁再編寫一些前端語言做美化。

**第7章 執行進度**

表7.1 專題進度表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **時間**  **工作項目** | **2月** | **3月** | **4月** | **5月** | **6月** | **7月** | **8月** | **9月** | **10月** | **11月** |
| **選定題目** |  |  |  |  |  |  |  |  |  |  |
| **擬訂計畫** |  |  |  |  |  |  |  |  |  |  |
| **找尋參考文獻** |  |  |  |  |  |  |  |  |  |  |
| **學習Python和OpenCV的用法** |  |  |  |  |  |  |  |  |  |  |
| **完成車牌辨識部分** |  |  |  |  |  |  |  |  |  |  |
| **完成網頁與資料庫部分** |  |  |  |  |  |  |  |  |  |  |
| **完成報告書** |  |  |  |  |  |  |  |  |  |  |
| **初步完成整個系統** |  |  |  |  |  |  |  |  |  |  |
| **測試並優化整個系統** |  |  |  |  |  |  |  |  |  |  |
| **完成整個專題** |  |  |  |  |  |  |  |  |  |  |
| **完成論文** |  |  |  |  |  |  |  |  |  |  |

**參考文獻**

1. Wikipedia，“影像處理”

https://zh.wikipedia.org/wiki/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86

1. 潘家鵬，林天斌，柯偉基，「車牌定位」，逢甲大學，資訊工程系專題報告
2. Wikipedia，“灰階” ，

https://zh.m.wikipedia.org/zh-tw/%E7%81%B0%E5%BA%A6%E5%9B%BE%E5%83%8F

1. 蘇彥彰，「車牌辨識系統」，國立聯合大學，資學院金腦獎參賽作品
2. Wikipedia，“二值化”，

https://zh.wikipedia.org/wiki/%E4%BA%8C%E5%80%BC%E5%8C%96

1. 楊茗富，鄭維勳，黃彥棖，陳俊宏，「車牌辨識系統」，崑山科技大學，資訊工程系專題報告 2013
2. Wikipedia，“OpenCV”，https://zh.wikipedia.org/wiki/OpenCV
3. Wikipedia，“Python”，[https://zh.wikipedia.org/wiki/Python](https://zh.wikipedia.org/wiki/Python2019/05/22)
4. 阮一峰的网络日志，高斯模糊的算法，

<http://www.ruanyifeng.com/blog/2012/11/gaussian_blur.html>

1. Python+OpenCV实现图像高斯模糊，

<https://blog.csdn.net/solidcorr/article/details/50625307>

1. 中值滤波器（Median filter）特性及其实现，

<https://blog.csdn.net/liyuanbhu/article/details/48502005>

1. 阿洲的程式教學，OpenCV教學，<http://monkeycoding.com/?page_id=12>
2. OpenCV-Python教程（11、轮廓检测），

https://blog.csdn.net/sunny2038/article/details/12889059

1. Wikipedia，“光學字元辨識”，

<https://zh.wikipedia.org/wiki/%E5%85%89%E5%AD%A6%E5%AD%97%E7%AC%A6%E8%AF%86%E5%88%AB>