

## 壹、前言

隨著城市交通擁擠問題的加劇，智能化停車場管理系統成為提高停車效率和改善用戶體驗的重要途徑。本研究聚焦於停車場門口的空位報位系統和車牌辨識系統的結合，旨在深入瞭解這一整合方案對停車場管理的影響。在傳統停車場管理方式面臨日益複雜需求的同時，我們探討了空位報位系統的不足之處，包括信息更新滯後和不靈活等問題。為解決這些問題，本研究引入了車牌辨識技術，以實現對停車場內車輛的實時監控和更準確的空位報位。車牌辨識系統的自動化和高度準確性為停車場管理帶來了新的可能性。通過與空位報位系統的結合，我們能夠實現即時更新的停車位信息，提供司機更靈活和準確的導引，減少徘徊時間，進一步提升停車效率。本研究綜合了現有文獻的評估和實證研究，分析了這一整合方案在提升停車場管理效能、減少交通擁擠以及提升用戶體驗方面的潛在效益。

### 一、研究動機

在現代的社會中，停車位不足已成為都市裡常見的問題，這不僅影響了交通流暢度，也給市民帶來了諸多不便。同時，車牌辨識技術已趨於成熟，這為解決停車方面的問題提供更多的選項，但在車滿為患的停車場，常常進去都找不到車位，儘管在門口都會顯示剩餘車位，但還是會遇到尚有空車位卻難以找到停車位的問題，因此我們想透過這次的專題在停車場門口的車牌辨識基礎上再加上顯示空車位的功能，我們決定要將此項議題搭配近幾年興起的 AI 技術和視訊鏡頭的輔助，製作一個自動車牌辨識及帶位功能的系統，幫助人們能更快且精準的在停車場找到可以停的車位。因此，本論文旨在探討車牌辨識及停車場帶位功能的應用，並確定其在改善城市交通管理和提升停車場效率方面的潛在價值。

### 二、研究目的

- (一) 評估各類型車牌辨識技術的準確性和效率，選出最合適的車牌辨識模型
- (二) 設計停車場空位管理以及報位功能的系統
- (三) 評估車牌辨識系統結合空位管理系統的效益和可行性

## 貳、文獻探討

### 一、自動車牌辨識系統

**自動車牌辨識 (Automatic License Plate Recognition, ALPR)** 透過攝像機捕捉彩色、黑白或紅外線等不同類型的影像，運用物體檢測、影像處理和模式識別等技術，以識別車輛的車牌號碼 (Du et al, 2012)。車牌辨識系統能應用於智慧城市的車輛管理、竊盜車輛調查、犯罪車輛追蹤和交通監控 (許博翔, 2021)。近年來車牌辨識技術已經非常成熟的應用於智慧停車場、交通收費系統等場域，然而運用於路口監視器影響，會面臨諸多挑戰

(謝亦信, 2019)。透過不同的自動車牌辨識，系統可以有效地實現車輛識別和監管，提升交通管理的效率和準確性。如下表一所示，車牌辨識模型比較表。

表一：車牌辨識模型比較表

車牌辨識方式	描述
基於影像處理的車牌辨識	利用電腦視覺技術，包括輪廓檢測、顏色過濾、二值化等步驟，對車牌進行分割和識別。適用於光線條件變化較小的場景。
基於深度學習的車牌辨識	利用深度學習技術，特別是卷積神經網絡（CNN），對車牌進行更靈活的辨識，能夠應對各種複雜的光線和角度變化，提高辨識準確性。
使用車牌識別硬體的車牌辨識	利用高解析度攝像頭和特殊光源等硬體設備，提高車牌辨識的效果。常見於停車場管理系統和交通監控系統。
基於車牌特徵的模型訓練	透過收集大量標記車牌的數據，訓練模型識別車牌特有的形狀和結構，從而實現準確的車牌辨識。
基於區域特徵的車牌辨識	考慮車牌在圖像中的相對位置和區域特徵，從而實現對車牌的快速而準確的辨識。

表一資料來源：研究者自行彙整分析

## 二、車牌辨識流程

**車牌辨識系統包含三個階段：車牌偵測、字元分割，與字元辨識（林永鑫，2018）。**系統通過從車輛進行的影像或視訊中擷取車牌圖像，再通過圖像處理和字元識別算法進行分析和識別，最終獲得車牌號碼。車牌辨識技術在各行各業中都有廣泛應用，例如停車場管理、交通違規監控等等。這些系統提高了處理效率和準確性，同時也減少了人力成本。透過自動化的辨識流程，車牌辨識系統能夠實現快速、準確地識別車牌信息，提升了管理效率和安全性。而其運作流程如下表二所示。

表二：車牌辨識流程

運作流程	說明
影像擷取	使用攝像頭捕捉車輛圖像，確保清晰的車牌區域。
前處理	對影像進行處理，包括去除雜訊、調整光線，以提高後續處理的準確性。
特徵提取	利用影像處理技術，提取車牌的特徵，如車牌的輪廓、字符的形狀等。
字符識別	使用模式識別技術，對提取的字符進行辨識，識別車牌號碼。

後處理	對識別結果進行後續處理，包括校正字符順序、錯誤修復等，提高準確性。
-----	-----------------------------------

表二資料來源：研究者自行彙整分析

### 三、Visual Studio Code 編譯器

Visual Studio Code (簡稱 VS code) 是由微軟於 2015 年推出的跨平台原始碼編輯器，支援多種程式語言的開發。與其他編輯器相比，它具有內建的擴充元件商店，使得使用者能夠安裝各種擴充元件，從而更方便地進行程式開發。在本研究中，我們選擇使用 VS Code 作為開發環境，用於車牌辨識系統和空位管理系統的編譯。如表三所示，各種編譯器比較表，我們進行了各種編輯器的比較，最終確定了 VS Code 的適用性。

表三：編譯器比較表

編譯器	特點
Visual Studio Code	<ol style="list-style-type: none"> <li>1.免費</li> <li>2.支援大量擴充插件的安裝</li> <li>3.幾乎支援所有語言的編譯</li> <li>4.跨平台編譯器</li> <li>5.開源</li> </ol>
Python 原生 IDE	<ol style="list-style-type: none"> <li>1.免費</li> <li>2.安裝 Python 附帶的編譯器，不須額外設定安裝</li> </ol>
Jupyter	<ol style="list-style-type: none"> <li>1.免費</li> <li>2.利用瀏覽器作為介面</li> <li>3.可以條列是直譯程式</li> <li>4.數據可視化</li> </ol>
Colab	<ol style="list-style-type: none"> <li>1.免費/付費</li> <li>2.可以連結雲端硬碟</li> <li>3.提供免費 GPU 資源</li> </ol>

表三資料來源：研究者自行彙整分析

### 四、語音播報系統

語音播報系統是一種通過語音播報訊息的技術，廣泛應用於公共場所、交通工具和智慧家電等領域。其中，AI 語音系統利用人工智慧技術生成自然流暢的語音，具有豐富的交互性和情感傳達能力，使得使用者能夠與系統更加自然地互動。然而，AI 語音系統需要大量的訓練數據和計算資源，且成本較高，對技術支援的要求也較高。

相對地，LED 跑馬燈是一種簡單且成本較低的播報系統，通常用於顯示簡單的文字信息。它易於安裝和使用，且在特定情境下具有一定的效果，如公共場所的公告和交通指示。

然而，LED 跑馬燈的顯示能力有限，無法傳達豐富的情感和交互性，且對於長篇訊息的顯示不夠有效。

總的來說，選擇適合的播報系統需根據具體情境和需求進行考量。如果訊息需要更豐富的交互和情感傳達，則 AI 語音系統可能更適合；而如果訊息簡單且不需要立即回應，則 LED 跑馬燈可能更實用。如表四所示，為 AI 語音與 LED 跑馬燈的比較表。

表四：AI 語音 vs. LED 跑馬燈比較表

	AI 語音	LED 跑馬燈
優勢	<ul style="list-style-type: none"> <li>●自然交流：AI 語音提供了自然的語音交流，更容易被人理解和接受。</li> <li>●多模式輸出：AI 語音可以提供語音和文字輸出，以適應不同需求的用戶。</li> <li>●聲音情感：AI 語音可以表達情感，更生動地傳達訊息。</li> </ul>	<ul style="list-style-type: none"> <li>●靜態信息呈現：LED 跑馬燈是適合呈現靜態訊息，如新聞標題或簡單通知。</li> <li>●不打擾：它不需要用戶主動觸發，可以輕鬆提供信息而不打擾用戶。</li> <li>●差異和應用情境：選擇 AI 語音還是 LED 跑馬燈應取決於訊息的性質和目標。</li> </ul>
劣勢	<ul style="list-style-type: none"> <li>●技術複雜，需要相對高的投資。</li> <li>●可能受到環境噪音的干擾。</li> </ul>	<ul style="list-style-type: none"> <li>●缺乏動態表達能力，信息傳達有限。</li> <li>●適用範圍受限，不適合複雜場景。</li> </ul>

表四資料來源：研究者自行彙整分析

## 參、研究方法

### 一、研究方法

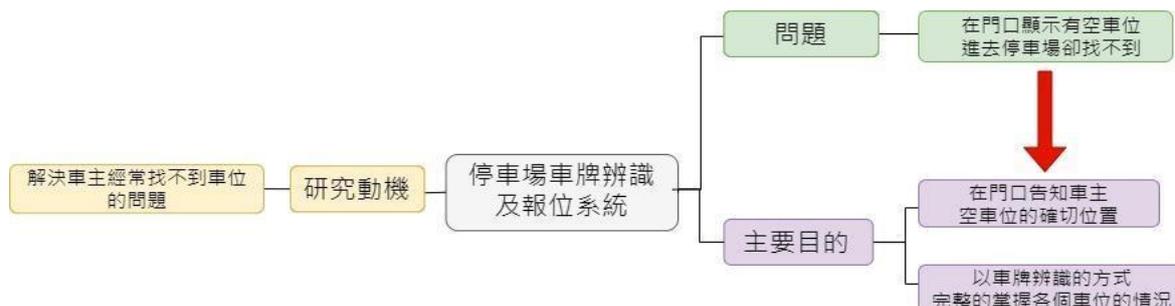
- (一) 文獻分析法：透過搜集相關領域資訊、論文、系統，深入了解這方面技術的應用和發展。
- (二) 實作研究法：對於我們所做出的自動車牌辨識及帶位功能的系統進行實測模擬，整理結果並進行分析改良。

### 二、研究架構

首先，我們計劃將車牌辨識技術與停車場的空位管理系統相結合，以清楚掌握停車場內車位的狀況。這項研究旨在解決車主在停車場門口看到空車位統計數量與實際情況不符的問題。我們計劃在停車場門口透過廣播方式告知車主空車位的確切位置，從而避免車主進入停車場後難以找到空車位的情況。我們的研究架構如下圖一所示，旨在通過結合車牌辨識技術和空位管理系統來提高停車場的運作效率，為車主提供更好的停車體驗。透過此項研究，我們希望能夠解決停車場管理中存在的實際問題，並為使用者提供更方便的停車服務，並進一步了解車牌辨識技術和空位管理系統的原理和應用。

## 停車場 waiter 讓你不再 wait

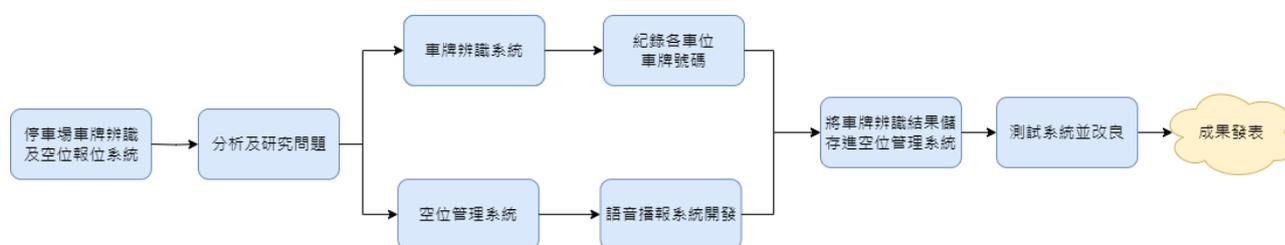
圖一：研究架構圖



圖一資料來源：研究者使用 draw.io 自行繪製

## 三、研究流程

圖二：研究流程圖

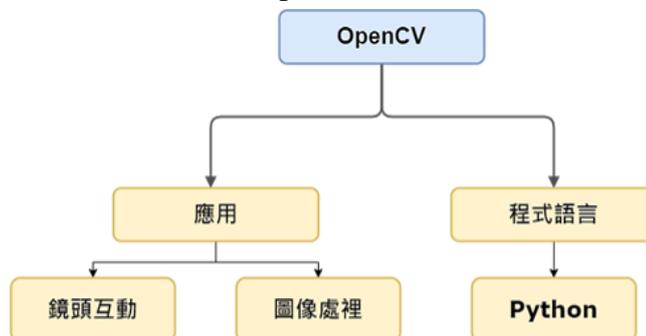


圖二資料來源：研究者使用 draw.io 自行繪製

## 四、研究設備與器材

- (一) 軟體環境：Visual Studio Code 編譯器、Python 軟體、Python OpenCV 套件，OpenCV 是一個跨平台圖像開源函式庫，能用於圖像和影片處理，Python 則是一款直譯式的程式語言，能用於 OpenCV 的開發，在本次研究中我們用以編寫車牌辨識和車位管理系統的程式。如下圖三所示，為 OpenCV 軟體套件。

圖三：OpenCV 軟體套件



圖三資料來源：研究者使用 draw.io 自行繪製

- (二) 硬體環境：

圖四：筆記型電腦



圖四資料來源：燦坤購物網，  
<https://myppt.cc/cpD8A>

圖五：1080P 視訊鏡頭



圖五資料來源：燦坤購物網，  
<https://myppt.cc/8lFTwy>

## 肆、研究分析與結果

### 一、車牌辨識系統實作

#### (一) 匯入函式庫

- 1、cv2 為 openCV，是一種跨平台的電腦圖形視覺庫，可以用於開發圖像處理、電腦視覺以及圖型識別程式
- 2、imutils 為基於 openCV2 的擴充元件，可以進行平移，旋轉，縮放等
- 3、numpy 為 python 相關的數字運算模組
- 4、Pytesseract 是 Google's Tesseract-OCR 的 python 封裝版，可以進行圖像中的字元辨識及處理。
- 5、playsound 是用以播放聲音的函式庫

圖六：匯入相關函式庫之程式

```
import cv2
import random
import imutils
import numpy as np
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r"C:\\Program Files\\Tesseract-OCR\\tesseract.exe"
import playsound
from playsound import playsound
```

圖六資料來源：研究者由 Visual Studio Code 實作截圖

#### (二) 圖片預處理

- 1、調整大小：首先，我們將輸入的圖片調整為所需的大小，以便後續處理。這有助於統一輸入圖像的尺寸。

- 2、灰階轉換：將調整大小後的圖片轉換成灰階圖像。這樣做有助於簡化圖像處理，因為在灰階圖像中，每個像素只有亮度信息而無色彩信息。
- 3、雙邊濾波：使用雙邊濾波器去除圖像中的噪聲，保留邊緣信息。這有助於提高後續邊緣檢測的準確性。

圖七：進行圖像大小及灰階去噪處理之程式

```
img = cv2.imread('D:\\coder\\ai car\\nxk8888.jpg',cv2.IMREAD_COLOR)
img = cv2.resize(img, (600,400) )
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = cv2.bilateralFilter(gray, 13, 15, 15)
```

圖七資料來源：研究者由 Visual Studio Code 實作截圖

### (三) 邊緣檢測及輪廓分析

- 1、Canny edge 檢測：採用 Canny 邊緣檢測算法，從去噪後的圖像中獲取邊緣信息。這將有助於找到圖像中的物體輪廓。
- 2、尋找輪廓：利用 Canny edge 取得圖形輪廓，並且在圖形上搜尋由大到小且密閉的 10 個圖形輪廓。這將有助於找到可能是車牌的區域。
- 3、篩選四邊形：由於車牌必定是四邊形，因此去除密閉圖形中非四邊形的部分。
- 4、標記車牌區域：找到正確的車牌區域後，在原始圖像上繪製一個紅框以標記車牌的位置。
- 5、遮蔽非車牌區域：將非車牌的部分進行遮蔽，以便更好地專注於車牌的識別。

圖八：像邊緣檢測及不相關部分遮蔽處理之程式

```
edged = cv2.Canny(gray, 30, 200)
contours = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(contours)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:10]
screenCnt = None

for c in contours:
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * peri, True)

    if len(approx) == 4:
        screenCnt = approx
        break

if screenCnt is None:
    detected = 0
    print ("No contour detected")
else:
    detected = 1

if detected == 1:
    cv2.drawContours(img, [screenCnt], -1, (0, 0, 255), 3)

mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(img,img,mask=mask)
```

圖八資料來源：研究者由 Visual Studio Code 實作截圖

#### (四) 車牌裁切與文字辨識

- 1、裁切車牌：利用標記的車牌區域，對原始圖像進行裁切，獲取車牌的部分。
- 2、文字辨識：對裁切後的車牌圖像進行文字辨識，以提取車牌上的文字信息。

圖九：裁切文字之程式

```
(x, y) = np.where(mask == 255)
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]
```

圖九資料來源：研究者由 Visual Studio Code 實作截圖

#### (五) 顯示結果

顯示文字：最後將辨識出的文字信息顯示在介面上，以完成整個車牌辨識的流程。

圖十：裁切文字之程式

```
text = pytesseract.image_to_string(Cropped, config='--psm 11')
print("programming_fever's License Plate Recognition\n")
print("Your license Number is:",text)
```

圖十資料來源：研究者由 Visual Studio Code 實作截圖

## 二、車位管理系統實作

- (一) 車位管理系統：儲存車牌號碼之位置及告知一個空車位。透過此系統，使用者可以快速找到可用的停車位，節省時間與精力。此系統不僅提高了停車效率，也有助於管理停車場的運作。此外，車位管理系統還能夠進行車輛進出的記錄，提高了安全性與管理效率。
- (二) 語音報位系統：利用 `playsound` 函式撥放 AI 生成之語音，使得報位更加直觀、生動。這項技術將 AI 生成的文字轉換為語音，提高了系統的易用性和可讀性，同時也為使用者提供了更加便捷的操作體驗。

圖十一：車位管理系統之程式

```

vancant_car = [[0]*5 for i in range(3)]
for i in range(3):
    for j in range(5):
        vancant_car[i][j] = False

vancant_license = [[0]*5 for i in range(3)]
for i in range(3):
    for j in range(5):
        vancant_license[i][j] = str('none')

parkingSpaceName = [[0]*5 for i in range(3)]
for i in range(5):
    parkingSpaceName[0][i] = str("A - " + str(i+1))
    parkingSpaceName[1][i] = str("B - " + str(i+1))
    parkingSpaceName[2][i] = str("C - " + str(i+1))
    
```

圖十一資料來源：研究者由 Visual Studio Code 實作截圖

### 三、針對系統進行 SWOT 分析

表五：系統之 SWOT 分析

Strengths 優勢	Weekness 劣勢
1.提升停車體驗：語音報位系統提供直觀、人性化的停車信息，提升用戶體驗。 2.節省尋找時間：快速告知空車位位置，減少車主尋找停車位的時間。	1.技術不確定性：語音辨識技術可能因環境噪聲、口音等因素而存在不確定性，可能導致誤報或未能正確報位。 2.成本與部署：實施語音報位系統需要額外的硬體和軟體投資，以及相應的維護成本。
Opportunities 機會	Threats 威脅
1.市場需求提升：人們對於停車便利性的需求不斷增加，提供更智能的停車場管理系統將受到市場歡迎。 2.整合其他服務：與移動應用程式、導航系統等整合，擴大應用範圍。	1.用戶接受度：一些用戶可能對新的報位系統持懷疑態度，用戶接受度可能成為威脅。 2.技術演進：技術的快速演進可能導致現有系統的過時，需要及時跟進升級。

表五資料來源：研究者自行彙整分析

### 伍、研究結論與建議

#### 一、結論

- (一) 本研究車牌辨識、空車位管理系統以及停車場門口的語音報空車位系統的整合，為現代城市交通管理和停車體驗帶來了新的可能性。這三個系統的結合不僅提高了停車場的運營效率，也為車主提供了更便捷、智慧化的停車體驗。
- (二) 車牌辨識系統的研究充分發揮了 AI 視覺和深度學習的優勢，達到了對車牌的高效辨識。透過車輛偵測、車牌定位、車牌分割和車牌辨識等步驟，本系統可以準確、快速地辨識進入停車場的車輛，並為空車位管理系統建立基礎資料。
- (三) 空車位管理系統的建立和互動介面的設計使停車場管理者和車主能夠實時了解停車場的情況。透過感應器和資料庫管理，該系統不僅提供了車位的即時佔用情況，還能透過網頁或手機應用程式達成車位查詢、預訂等功能，提高了停車場的運營效益。
- (四) 在停車場門口的語音報空車位系統進一步提升了用戶體驗。透過整合語音報告技術，車主能夠在進入停車場的同時即時獲取空車位的信息，無需長時間尋找停車位，能有效減少車主的等待時間，提高了整體停車體驗。

## 二、建議

- (一) 提供選擇停車位服務，例如是否需要充電車位及身心上愛車位等。
- (二) 建立一個更即時性的管理系統，若有車主並未按照給予的車位停車，能及時的修正此項問題。

## 陸、參考文獻

- Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2012). Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Transactions on circuits and systems for video technology*, 23(2), 311-325.
- 許博翔(2021)。輕量化車牌辨識模型。國立臺灣師範大學電機工程學研究所：碩士論文。
- 謝亦信(2019)。以深度學習架構實現之適用於複雜道路環境之車牌辨識系統。國立臺灣師範大學電機工程學研究所：碩士論文。
- 林永鑫(2017)。植基於卷積神經網路之高效能車牌辨識系統。國立臺灣師範大學電機工程學研究所：碩士論文。