

中華民國第 64 屆中小學科學展覽會
作品說明書

國小組 生活與應用科學(一)科

第三名

082813

百密一「疏」～校園智慧疏散系統之建構

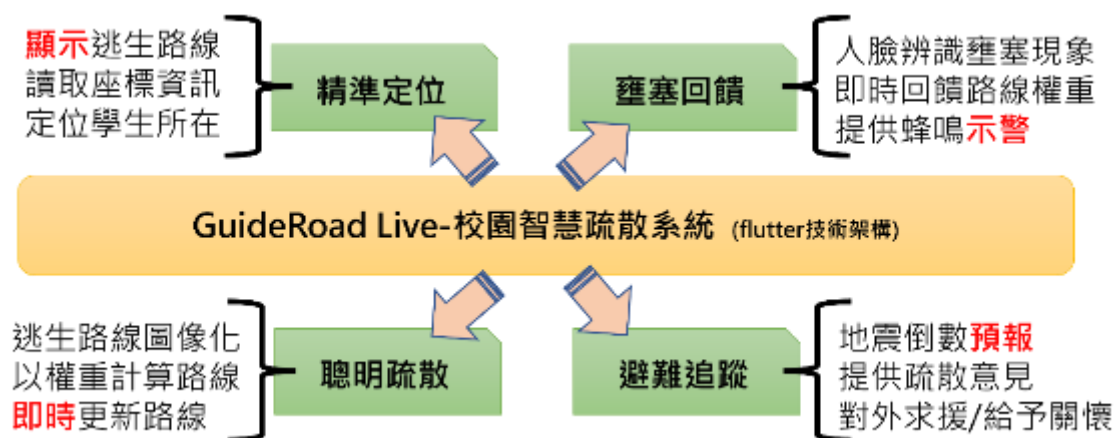
學校名稱： 新北市板橋區沙崙國民小學

作者： 小六 鄭雨萌 小五 王侑希	指導老師： 王亭雅 林建毅
---------------------------------	-----------------------------

關鍵詞： Dijkstra 演算法、校園疏散

摘要

此研究建構一套校園智慧疏散系統—GuideRoad Live，以協助學校規劃逃生路線，並在緊急狀況下迅速提供疏散建議。該系統具備精準定位、壅塞回饋、避難追蹤、聰明疏散等四大功能。在**精準定位**方面，系統進行地圖建模，結合校園 WiFi 數據庫，分析教室 GPS 資料，實現個人定位。在**壅塞回饋**方面，系統透過監視器分析樓梯壅塞情況，即時回傳數據、更新路線，並設計蜂鳴器警報，降低學生恐慌與推擠風險。在**避難追蹤**方面，系統匯入地震級數，提供即時警示預估逃生時間，提供逃或躲的疏散意見，並對外援助與給予使用者關懷。最後，系統根據節點、邊、人流速度、壅塞回饋的即時數據等因素進行權重計算，達到**聰明疏散**的目標。



(圖片由第二作者繪製)

壹、前言

一、研究動機

1999 年發生的 921 大地震後，學校在每年 9 月 21 日都會進行地震演練，要求全校師生在短時間內疏散至操場，聽老師們說學生很容易擠在同一個路口難以前進，了解後才得知目前校園疏散路線是以平均分配的方式設計，忽略了人多的密集度與實際動態因素的影響，我們不禁思考這些路線真的對學生而言是最快且最安全的嗎？地震來時，到底要趕快躲起來還是要逃跑才可以活命呢？如果某個樓梯臨時產生人潮擁擠，能夠即時知道可以往其他路線疏散嗎？因此我們決定規劃一個可以提供安全疏散資訊的系統。

查詢相關文獻，大部分都聚焦於火災防災路線設計，以及校園逃生出口空間大小分析，對於應對地震等緊急情況的校園疏散相關討論反而不多。在今年度 0403 大地震後，更有不少專家學者呼籲老舊校舍的安全性，這也是目前必須被重視的潛在問題；老師們曾分享 SDGs 目標 11「建構具包容、安全、韌性及永續特質的城市與鄉村」，使城市和人類居住更加安全、可持續性。為了讓校園安全的控管提供一個更全面的思考。讓校園應對未來可能遇到的各類緊急情況，實現更安全、更可持續的校園環境，更讓我們意識到建構出一個屬於校園的智慧疏散系統勢在必行。

二、研究目的與研究問題

我們研究自校園地圖程式化，演算法分析最佳路徑，以 Flutter 技術設計定位、偵測、回饋與互動，建立校園智慧疏散系統- GuideRoad Live，訂定以下研究目的與研究問題：

（一）資料調查與校園地圖建模

1. 校園中有哪些關鍵地點（如教室、出口、樓梯等）需要在地圖中表示？
2. 校園地圖應如何表示與連接關鍵地點？
3. 如何利用 python 語法進行地圖建模，將校舍環境程式化？

（二）建立最佳疏散系統模型

1. 如何依據演算法計算各項權重列出最佳路線？
2. 分析與比較不同因素類型，以及這些因素如何影響疏散系統？
3. 實測不同年級、數量學生在不同路線上，對疏散結果有何影響？

（三）建立跨平台、可移動式的校園智慧疏散系統

1. 如何以 flutter 的技術架構進行個人定位與建立校園疏散的必要資訊？
2. 如何納入影像偵測、即時回饋與進行互動，建立系統邏輯？
3. 如何加入求援與情感機制，建立校園智慧疏散系統- GuideRoad Live？

三、文獻探討

(一) 名詞定義：

1. **最佳路徑**：綜合多項因素（如：時間、人流量、建物安全性）後找出最好的路線。研究初期規劃的為即時疏散路線，因此考慮實際校內路隊行走時的路線，有時似乎是最短的距離，但是有可能因為人數太多、轉彎比較多等因素，不見得可以最快抵達目的地，因此最後定義我們研究目的為找出最快能夠抵達終點的路線，為考慮多項因素的最佳路徑。
2. **權重**：可以視為計算路徑時經過一條邊所需的成本，在研究中是指 **Dijkstra 演算法** 中評估各項疏散路徑因素所計算出的校正數值。

(二) 校園疏散地圖：

在 921 地震後，校園每年都需針對地震進行避難疏散演練，也因此各校皆有針對校園班級的疏散路線（圖 1-1），繪製各校的校園疏散地圖，我們分析現行學校疏散地圖條件如下：



圖 1-1 防災疏散路徑圖
(照片由第一作者拍攝)

1. 疏散人流皆匯集前往操場，方便清查人數。
2. 導致部分班級會在疏散逃離建物後，重新穿越另一座建物作為疏散路線。
3. 疏散皆以每層樓分兩邊前進。並無考慮年級間速度不同可能導致的壅塞現象。
4. 無訂立疏散時遇到壅塞情況下的反應條件或疏導模式。

(三) 逃生相關文獻：

1. 內政部築研究報告「學童避難速度調查研究」則是對於國小學童在校內避難時的逃生速度進行探討。研究調查各種因素對於學生行走速度的影響包含：
 - (1) 學童快走水平距離分別以 10m、25m 和 50m 三種距離進行比較，發現 25m 和 50m 的速度是有差異的，此外不同年級身高體重皆會對於速度造成影響。
 - (2) 學童水平逃生速度會受到年級身高和體重影響，而垂直速度的結果則發現下樓時 4-3 樓和 2-1 樓最快，3-2 樓則最慢。
 - (3) 不同門寬所造成的通道瓶頸反應不同，十秒內人數小於 36 人，150 cm 門寬，人流呈現較為順暢。

2. 沈宸葳(2017)在「智慧型室內防災疏散系統之設計與規劃」中，將室內空間簡化成簡易路線圖（圖 1-2），並利用類神經網絡進行演算搭配各種感測器，找出最佳逃生路線。

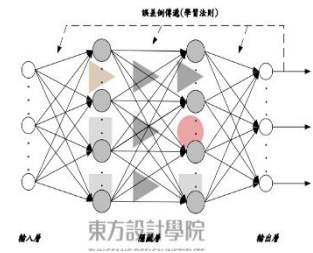
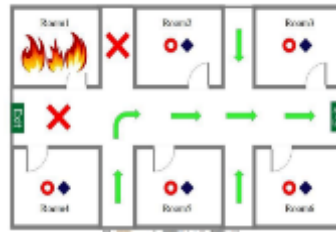


圖 1-2 逃生簡易路線圖
(圖片來源: 沈宸葳(2017))

3. 縣市工務局有提供建物套繪地籍圖資料，查閱校內目前建物的建造年份圖（圖 1-3），多為民國 68 年至 80 年間所建造的老舊建物。依據中華民國內政部國土管理署公告之建築物耐震設計規範及解說，初版資料為 94 年 12 月 21 日公告，並自中華民國 95 年 1 月 1 日生效，顯示目前建物都在該法規設立前完成興建。



圖 1-3 學校的建造年份圖
(圖片來源: 工務局網站)

(四) 小結

由上述的文獻與相關資料，發現對於疏散的最短路徑有時不見得是**最快抵達目的地**的最佳路徑，必須同時考慮多重因素的影響，因此總結出幾項重點作為研究方向：

1. 不同年級學生的行走速度於不同路線（包含廊道、樓梯）的權重有很大的差異。
2. 調查校舍興建年限與法規限制，可作為疏散路線設計的權重考量項目。
3. 現行疏散地圖規劃單一個集合點，可方便點名但是實際執行時易造成壅塞，可思考增加逃生出口（或路線）的可行性。
4. 因應不同人流變化對於疏散產生的影響，我們認為應加入偵測、回饋、互動等條件，強化系統的「即時性」與「可調整性」。

貳、研究設備與工具

一、實驗設備：旗子、碼表、捲尺、膠帶、ESP32 驅動板、蜂鳴器

二、實驗工具：Jupyter Notebook、演算法路徑搜尋模擬器、ChatGPT4.0、Flutter、Arduino IDE 編譯器。

在實驗工具中，我們與老師嘗試透過 chatGPT，輔助我們完成程式的修正。在與 chatGPT 討論的過程中，我們發現「轉化問題、拆解知識和反覆驗證」是和 chatGPT 協

作的重要關鍵。為了有效討論出能夠執行的程式語法，我們反覆執行以下幾種不同問句方式讓 AI 不斷修正語法，其問句方式如表 2-1：

表 2-1 chatGPT 協作問句方式

方式一：具體描述執行的條件以及轉化（簡化）問題	
告知 AI 使用的編碼工具以及演算法，並列出我們希望建構的圖形需要滿足的條件。	
提問	回覆
我使用的工具是 anaconda 的 jupyter 編輯器，語言為 python，演算法為貪婪演算法。 程式邏輯中幾項條件： 1、一定要透過樓梯才能在樓層間移動 2、同一樓層可左右隨意移動 3、出口有左右不同邊，請比較左右出口後，再推薦最短路徑圖。我提供的區域矩陣如下： (以下省略)	
方式二：將複雜問題拆解成一個問題，執行成功後，再慢慢加入其他問題修正。	
提問	回覆
好，請讓我將問題簡化。 我將四個修改意見，改為 1 個：在原節點和權重設計不變的前提下，將程式修改為所有的 Room 節點同時往 Exit 移動。並且提供文字檔案的各節點至 Exit 的權重計算值。	
方式三：反覆拆解知識，確認各項語法功能補充需要的條件，請 AI 確認語法是否有達成，如果沒有完成 AI 會協助修正。	
提問	回覆
我已經完成了正確的程式碼。目前想要修改大樓的樣子，但新大樓的樓梯會遇到三方向往來的路徑，我該如何改寫矩陣呢？ 修正問題： 你誤解我的問題。我的問題是在同一樓層有三個方向的路線，無法以直線的方式撰，我該如何修改。	
方式四：反覆驗證，追問詳細作法	
補充需要的條件，請 AI 確認語法是否有達成，如果沒有完成 AI 會協助修正。	
提問	回覆
請逐行，詳細說明上述程式碼的意義。 請逐行解釋空的圖 G 的程式碼。 pos 內的數值是否代表 xy 座標的位置系統呢？	

參、研究過程或方法

一、研究架構圖

我們的研究可分為三個研究目的，此章將分成三節逐步介紹研究目的一、二、三的研究歷程與實驗方法，其研究架構圖 3-1 如下：

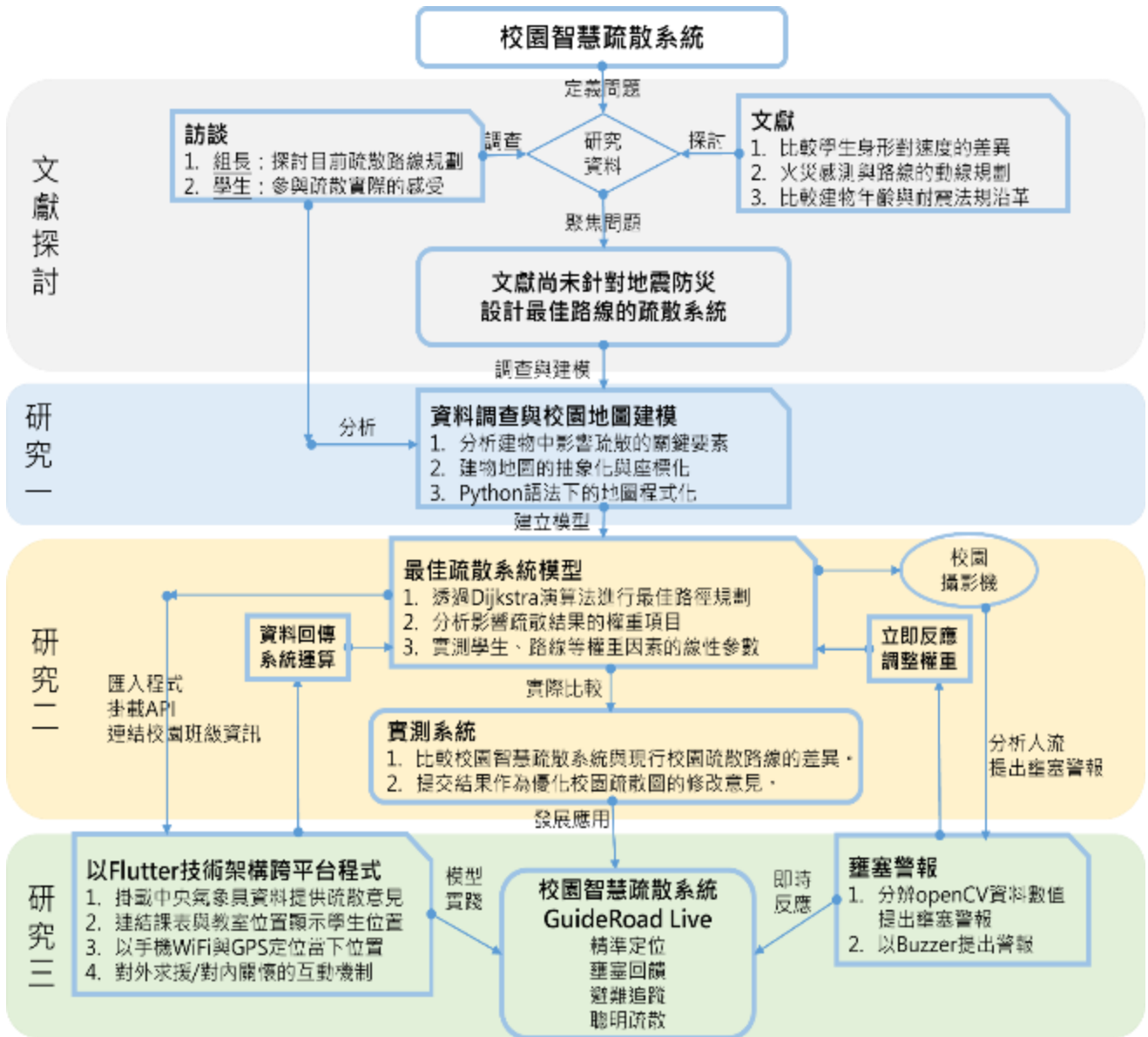


圖 3-1 研究架構圖

(圖片由第一作者與第二作者共同繪製)

二、資料調查與校園地圖建模

根據文獻中師長、學生訪談、校園監視器畫面資料調查，我們確立了教室、樓梯和出口為影響疏散系統的關鍵要素，因此透過建築物抽象化概念，將建築物轉變為節點圖，可以減少繪製複雜的地圖，其抽象化、座標化至程式化流程如下。

(一) 資料調查

1. 師長訪談

為了了解目前校園的疏散路線規劃方式，我們訪問了幾位學校有擔任生教組長經驗之教師（圖 3-2），了解各校如何進行逃生路線規劃，訪問對象與背景：三位現任生教組長，新北市板橋區兩位、新竹市一位。



圖 3-2 訪問組長圖片
(照片由第一指導老師拍攝)

2. 學生問卷調查

由於我們覺得可能有我們沒有想到會影響疏散速度的因素，因此參考文獻「學童避難速度調查研究」中幾項因素設計表單給同學填寫，想知道有哪些影響逃生速度的因素，表單如圖 3-3、3-4。

(1) 你認為以下哪些選項會影響疏散的速度？(複選)

- 出口大小
- 出口位置
- 出口容納人數
- 年級(人)
- 人數
- 樓梯
- 走廊寬度
- 轉彎(走廊)

(2) 早上跑，你覺得哪一項因素對走路速度的影響最大？

- 出口大小
- 出口位置
- 出口容納人數
- 年級(人)
- 人數
- 樓梯
- 走廊寬度
- 轉彎(走廊)

(3) 除了以上因素，請問你覺得還有什麼因素會影響疏散的速度？

圖 3-3 學生調查問卷問題
(圖片由第一作者截圖)

圖 3-4 調查表單
(圖片由第一作者截圖)

3. 校園錄影畫面比對

由「學童避難速度調查研究」文獻中得知，逃生出口寬度在十秒內通過人數小於 36 人，150 cm 門寬的情形下，是可以比較順暢不會造成擁擠的，我們量測到學校的樓梯寬度為 3.4m 至 3.5m 間，主要出入口寬度 1.6m 至 2.4m 範圍。我們觀察全校放學時的錄影畫面，比較主要路口，提供我們深入了解實際人多時的移動情形。放學路隊選定人潮較集中的主要樓梯口進行觀察，包含樓梯 A、C、E、F，位置（圖 3-5）。



圖 3-5 校園主要路口位置
(照片由第一作者拍攝)

(二) 轉化校園地圖

1. 抽象化

我們設定全校逃生演練的集合點作為目的地繪製單一樓層，首先進行校舍數據測量，接著完成全校三層樓梯的簡圖（圖 3-6），使各節點間未產生交叉，最後加入各樓層的教室節點，完成全校地圖的簡圖。

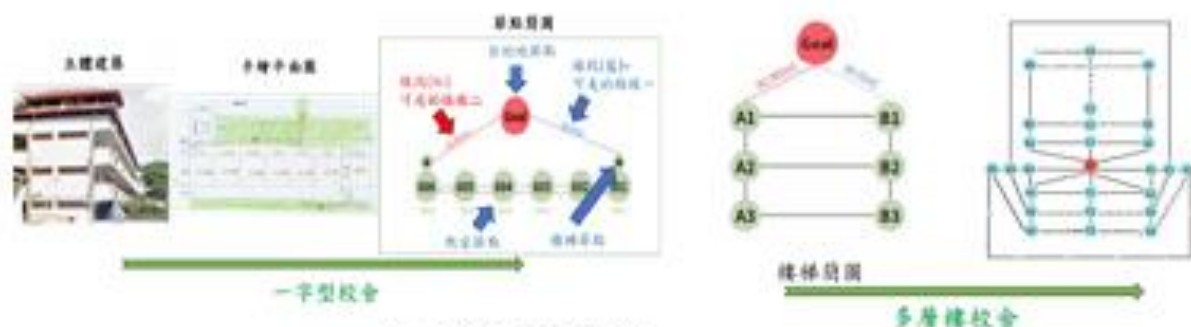


圖 3-6 校園地圖抽象化
(圖片由第一作者與第二作者共同拍攝與製作)

2. 座標化

進入程式編輯前，分別以手繪、Exce 表格方式，輔助我們定位複雜的地圖，以表格中的每一格代表一個位置定位（圖 3-7、3-8），標示校園節點的(x,y)轉化。



圖 3-7 校園地圖座標化手繪(照片由第一作者拍攝)



圖 3-8 校園地圖座標化 excel
(圖片由第二作者製作)

(三) 校園地圖程式化

為了將圖形轉換提供電腦進行判讀，我們分析兩種不同的程式建圖模式：“numpy”陣列（矩陣）表示和“networkx”圖形表示。根據兩種不同的城市建圖模式，我們分列比較如表 3-1，最後因建物需求，我們使用同一節點上可多點延伸的 networkx 圖形。

表 3-1 比較“numpy”陣列表示和 networkx 圖形表示(圖片由第二作者繪製)

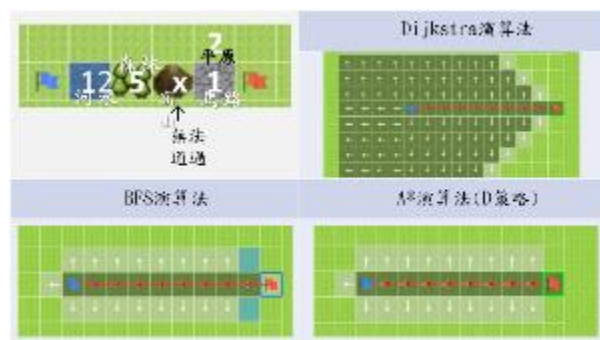
	numpy 陣列（矩陣）表示	networkx 圖形表示
差異	<ul style="list-style-type: none"> ●用於廣泛的數值計算。 ●抽象的數值矩陣形式，對於圖的結構主要以通過位置和數值來表達。 	<ul style="list-style-type: none"> ●專門用於處理和分析圖形數據，支持豐富的圖形理論算法和模型。 ●節點和邊可以帶有多種屬性。
簡圖		
結果	無法繪製複雜環境的大樓	可依據需求，擴充同一節點上，連結的節點和邊的數量，符合需求（採用）

三、建立最佳路線的系統模型

為考量研究中許多不同的因素可能造成路線規劃的影響，我們透過路徑搜尋模擬器認識不同的演算法，接著根據演算法需求列出討論的權重項目與分析，最後以實驗量測與資料分析比較這些因素對於疏散的影響，說明如下：

(一) 評估校園疏散的最佳演算法

路徑搜尋模擬器（圖 3-8）繪製三種不同的路徑地圖，比較 Dijkstra 演算法、A* 演算法兩種演算法的計算方式差異，透過模擬器提供的觸及格數、運算成本和路徑格數等數據進行比較。



<https://pathfindingdemo.gamelet.online/>

圖 3-8 模擬器演算法範例圖

(圖片由第一作者整理截自路徑搜尋模擬器)

(二) 權重的探討

研究依據 Dijkstra 演算法探討影響最佳路徑的各種權重，參考訪談、問卷、新聞、現況等條件，考量面向如表 3-2：

表 3-2 權重總表

權重項目	說明
人流速度	不同年級學生進行速度不同，應賦予不同年級學生不同疏散權重值。
走廊樓梯長寬	依據實測結果進行權重設計
班級特殊生	身體障礙或行為障礙學生的逃生速度和風險不同，
班級人數與位置	各年級各班學生人數不同，且因不同課程可能會有跑班現象，留意不同時間下班級人數與學生位置的變動性。
出口規劃	依據原始的地震防災路線，學校的疏散集合地點為操場位置。
疏散順序	不同樓層間疏散時可能產生壅塞現象，必須安排紓解壅塞的順序。
壅塞示警	當樓梯人數過多時，調整該樓梯權重，並進行壅塞示警。
建物年齡	依據建物年齡、耐震補強資料與耐震法規沿革等資料進行權重討論。

(三) 速度量測方法說明：

1. 參考文獻「學童避難速度調查研究」量測速度方法修正。兩人的工作分配

如圖 3-9 所示，實際量測步驟如表 3-3。

人員	工作項目	工具
測驗員A	喊預備-開始(同時揮旗) + 按碼表計時	碼表 旗子
測驗員B	按下碼表計時	碼表

圖 3-9 速度量測工作分配圖
(圖片由第二作者製作)

表 3-3 時間量測步驟說明

step	說明
0	準備階段 A實驗者(左邊藍色小人)一手拿平板一手拿旗子，B實驗者(右邊藍色小人)一手拿平板，站在第一的受實驗者(第一個黑色小人)腳對齊起點線(紅線)。
1	Start End 當A實驗者對準起點時第一個受實驗者就開始走，A和B實驗者同時按下開始鍵。
2	Start End 開始走之後A實驗者要跟在受實驗者後面。
3	Start End 當站在後面的受實驗者跨過終點線時，A和B實驗者同時按下時間停止鍵。

註：為了盡可能降低實驗誤差，實驗數據則會採用兩人的數據取平均

2. 人流速度的權重參數計算

- (1) **人數與時間的關係**：分別測試了 10、20、50 個人走同樣距離的走廊、樓梯和轉彎的時間，繪製出折線圖以了解此因素對於中年級學生行走時間的影響。
- (2) **年級與時間的關係**：對低、中、高年級學生(各 50 人)計時，分析不同年級學生的差異性。
- (3) **找出因素之間的關係方程式**：將數據透過 Excel 分別帶出線性方程式($y=ax+b$)，將速度轉化為權重，並且以 25 人作為基準。再找出不同人數所需的時間。
- (4) **「時間」作為權重參數**：使行走速度較快的學生獲得較低的權重，等於道路對他們的負荷較小，於是以時間作為每一個線段的權重參數。

3. 校園走廊、樓梯長寬的測量

諮詢學校相關人員後，發現由於學校建校已久，過去的建築資訊皆為紙本檔案，但經過多次改建也已無完整圖面資料，因此必須透過實際儀器人工量測(圖 3-10)。



圖 3-10 實際量測走廊長度
(照片由第一指導老師拍攝)

(四) 小結:

根據研究目的一選定 networkx 圖形地圖建模，此節以路徑搜尋模擬器決定 Dijkstra 演算法確定演算法基礎，分析不同的權重項目，並在最後說明研究中進行人流速度、走廊樓梯長寬等權重的研究方法。

四、建立跨平台、可移動式的校園即時疏散系統

(一) Flutter 技術架構 (軟體)

Flutter 是一個可以跨平台設計的框架，是 Google 開發的 dart 程式語言編寫讓 Android、Web 等不同平台所共用的技術架構操作介面

(圖 3-11)。我們利用

Flutter 協助製作 Android 與

Web 共用的程式，為了要串連不同程式語法與資料，將設計的系統規劃與操作方式以功能區分如表 3-4：

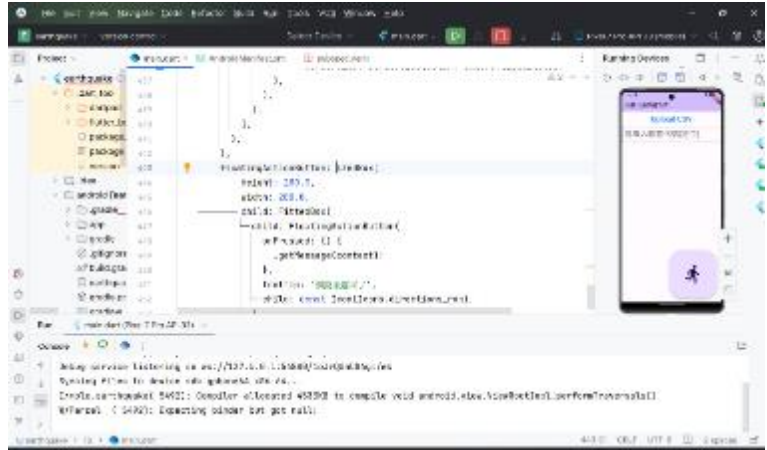


圖 3-11 Flutter 架構操作介面
(圖片由第一作者截圖自 flutter)

表 3-4 Flutter 系統功能與設計總表

功能	設計方式	連結工具
定位 教室	<ol style="list-style-type: none"> WiFi 定位範圍廣，並且校園已有全覆蓋 WiFi。其次，WiFi 定位低能耗也有足夠精準度。 建立出校園教室位置資料庫，使用 Geolocator 抓取所在位置的經緯度 (GPS) 後與資料庫連結，定位教室位置。 匯入班級人數與課表資料庫，結合研究結果二中：不同年級學生對人流速度影響的權重分析。 	Google Map Geolocator app 位置資料庫
連結 資訊	<ol style="list-style-type: none"> 取得中央氣象局的公開資訊的 API 與 JSON 資料檔，將資料序列化。使用 HTTP 套件連結 API 與 JSON 模板。 利用取得的資訊加入計算語法，定義系統所需要的參數：預估到達秒數 與 震度(PGA) 將獲得的參數預估到達秒數 與 震度(PGA) 匯入後與 Python 程式模型比對。 	中央氣象局氣象公開資訊-Flutter 套件- json_serializable 與 HTTP 地震資料庫

震央距離 =

$$\sqrt{((\text{地標經度}-\text{震央經度}\times-1)\times 101)^2 + ((\text{地標緯度}-\text{震央緯度}\times-1)\times 111)^2}$$
 震源距離 = 深度²+震央距離²
 震波已走距離 = (目前時間-地震發生時間) × 3.5
 剩餘秒數 (若為負表示 已抵達) = 震源距離-震波已走距離 ÷ 3.5

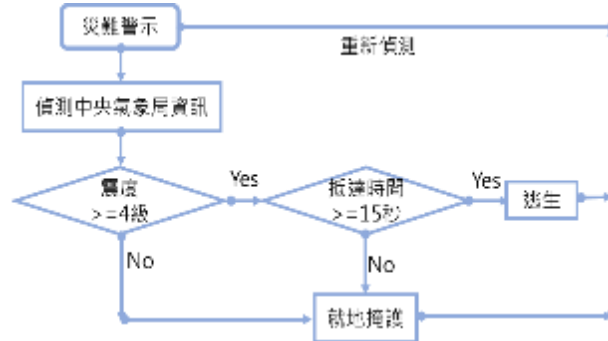
計算
路線

連結研究二中所建構出 Python 程式，設定為四個出入口，計算出抵達最近出口的時間以及路線，再與地震所得資料比對後，提供災難提醒機制。

Python

實際實驗量測避難時，語法邏輯為：

災難
警示



Flutter

(二) openCV 的影像偵測與壅塞回饋

openCV 函數庫具備相當多元的辨識系統，我們的研究相較於人臉辨識，更適合使用 Haar Cascade 上半身分類器：haarcascade_upperbody.xml，針對人體圖像的上半部分進行偵測。

我們將校園監視器位置訂於樓梯。根據偵測結果，我們給予壅塞權重的調整。考量學校班級學生多為 20 至 30 人為一班，因此討論，當有一個班級學生（約大於 20 人）排隊時，樓梯節點權重改為 2 倍，而當有兩個以上的班級學生（約大於 40 人）排隊時，樓梯節點權重改為 3 倍。並同時提供 ESP32 序列埠數值，進行警報作動，語法邏輯如圖 3-12。

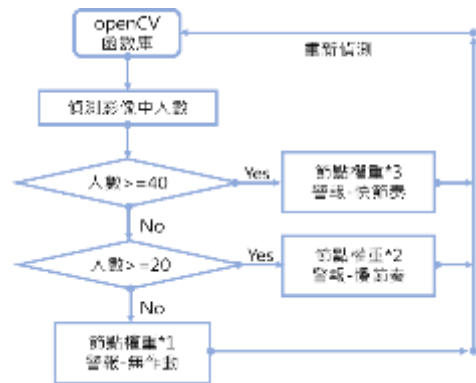


圖 3-12 openCV 邏輯架構
(圖片由第二作者繪製)

(三) 避難追蹤的求援與情感機制

想像地震發生後的受困情境，當地震發生後，若判斷地震大於 4 級且 5 分鐘後手機使用者沒有任何移動的反應，設定手機連結助理功能或自動發出訊息推播與使用者對話，對使用者進行關懷確認，語法邏輯如圖 3-13。

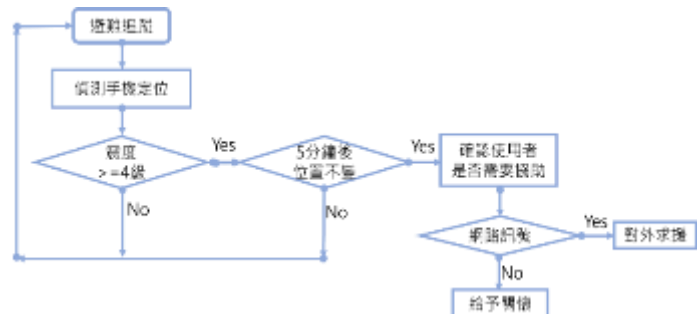


圖 3-13 求援與情感機制
圖片由第二作者繪製

肆、研究結果與討論

一、資料調查與校園地圖建模

(一) 資料調查

經過各項資料調查後，我們總結出幾個影響因素於表 4-1 中。

表 4-1 影響校園疏散因素總表

	出口大小	出口位置	走廊長寬	年級	人數	樓梯大小	路線轉彎處
組長	●		●	●	●	●	
問卷	●				●		
文獻	●		●	●	●	●	
監視器	無差異	無差異	無差異	◆	◆	◆	◆
實作				◆	◆	◆	◆

文獻討論與實作確認

1. 師長與學生資料蒐集（說明見 p.6）

表 4-2 師長與學生說明因素整理

對象	說明
組長 (3名)	<ol style="list-style-type: none"> 1. 年級的教室配置：高年級在高樓層、低年級排在低年級； 2. 特殊狀況的班級（走得會比較慢的）：班級教室安排在一樓。 3. 分配時會考慮到走廊寬度，思考能夠一次安排幾個班級同時行走。
學生問卷 (10名)	<p>第一名：出口大小與人數。</p> <p>第二名：出口位置與走道寬度</p>

2. 校園錄影畫面比對

透過監視器觀察週三大放學的情形(如表)，因此觀察的錄影畫面（表 4-3）主要為樓梯 A、C、E、北側門出口，由影片中發現，四個主要出口唯有樓梯 E 部分因為下樓後隨即連續有兩個轉彎，稍微有停滯情形發生，但是整體行進還是順暢的。

接著我們以監視器畫面計算每十秒通過人數比較各路口情形，以了解停滯行為是否對於路口通過人數帶來影響，四個路口的通過人數約 16 至 20 人，並不受出入口寬度的影響，而樓梯 E 部分的通過人數是最多的 20 人，顯示這路口可能因為人數較多而略顯緩慢（表 4-4），但實際上通過人流情況確實是順暢的，這些結果與文獻的結果對比是符合的，因此我們的實驗暫不考慮調查到的「出口大小」因素。

表 4-3 影片觀察紀錄表 (照片由第一作者拍攝)

位置:樓梯 A 寬度:1.6 公尺 時間:2024/2/21 1 描述:行走順暢保持間距, 不會有停滯或閃避情形。	位置:樓梯 C 寬度:2.2 公尺 時間:2024/2/21 描述:通過學生人數較少, 都是以班級為單位, 非常順暢。	位置:北側門出口 寬度:2.4 公尺(或更大) 時間:2024/2/21 描述:行走順暢保持間距, 不會有停滯或閃避情形。	位置:樓梯 E 寬度:2.1 公尺 時間:2024/2/21 描述:彎道過來位置學生會略有停滯, 但是速度順暢。
			

表 4-4 不同樓梯行走人數統計表

位置	次數	時間範圍	人數	平均
樓梯 A	1	12:08:01~12:08:11	19 人	16 人
	2	12:08:58~12:09:08	13 人	
樓梯 C	1	12:03:41~12:03:51	18 人	20 人
	2	12:01:03~12:01:13	22 人	
樓梯 E	1	12:07:00~12:07:10	16 人	19 人
	2	12:09:33~12:09:43	23 人	
北側門出口	1	12:06:26~12:06:36	19 人	18 人
	2	12:08:47~12:08:57	17 人	

(二) 轉化校園地圖：抽象化與座標化

為了分辨全校的樓梯，我們統一設定樓梯為編號 A~G，一字型部分有兩個樓梯 A、樓梯 B，不同層樓則是將樓梯編號加上樓層的數字表示，完成樓梯圖後再加入了樓梯間的所有教室編號，最後定義出一樓的多個彙集位置點以方便路線規劃，完成了全校簡圖的發展過程圖 4-1。

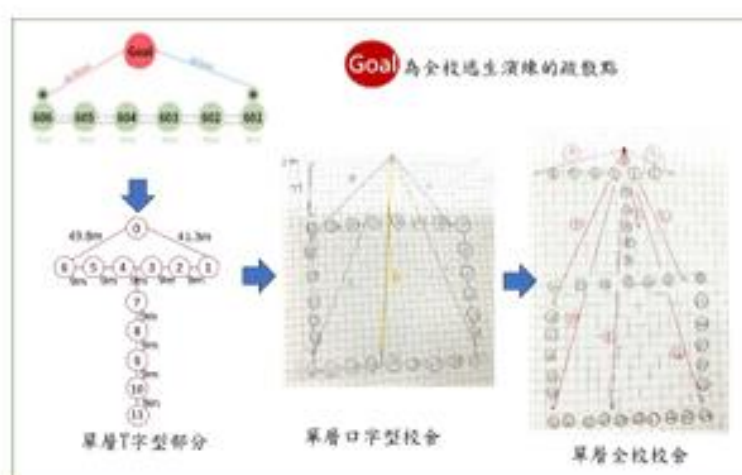


圖 4-1 校園地圖抽象化發展圖
(圖片由第二作者繪製)

(三) 校園地圖程式化

根據上述資料蒐集與分析的結果，我們將出口、樓梯、教室等條件作為節點(node)進行建模，並利用 networks 圖形上邊(edge)的資料連結各節點，逐步建構程式化地圖的發展如圖 4-2。

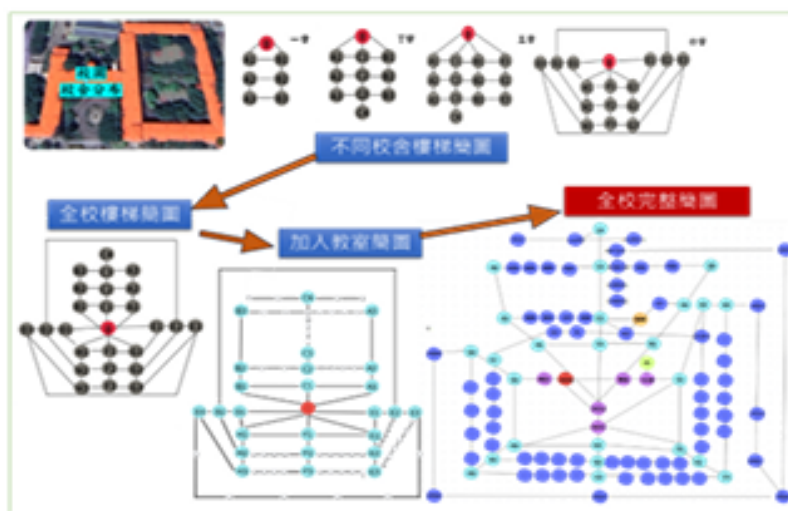


圖 4-2 校園地圖程式化發展圖
(圖片由第二作者繪製)

1. 地圖程式立體化

為了使圖形的觀看感更直覺更有逃生效果，進入程式設計時再修正三點：

果，進入程式設計時再修正三點：

- (1) 節點圖以「立體圖形」方式呈現。
- (2) 疏散路線的圖形加上「箭頭」。
- (3) 逃生出口修正為 4 個

(報告書 p.9 說明)。

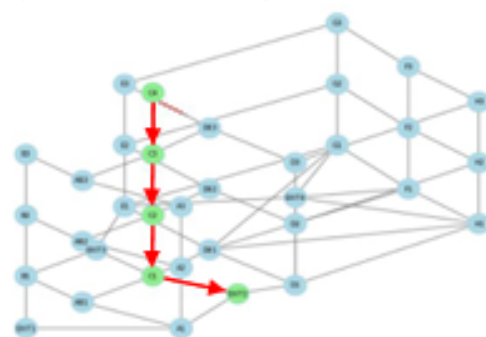


圖 4-3 校園地圖程式立體化
(圖片由第二作者繪製)

2. 部分具體的建模程式碼如表 4-5

表 4-5 校園地圖建模程式碼

類別	程式碼
node 節點實例	<p>G.add_node (節點名稱, 節點位置, 年級屬性, 人數權重)</p> <p>G.add_edge (節點 1, 節點 2 連接, 線段權重, 線段屬性)</p> <p>G.add_node("R608", pos=(7, 11), node_type='Y3', node_weight=1)</p> <p>G.add_node("R609", pos=(6, 11), node_type='Y3', node_weight=1)</p> <p>G.add_node("R508", pos=(5, 11), node_type='Y3', node_weight=1)</p>
edge 邊實例	<p>G.add_edge("R601", "B3", weight=5, type='room-edge')</p> <p>G.add_edge("R601", "R602", weight=5, type='room-edge')</p> <p>G.add_edge("A2", "A1", weight=5, type='staircase-edge')</p> <p>G.add_edge("C2", "C1", weight=5, type='staircase-edge')</p>

(四) 小結

此節定義研究中的地圖建模基礎，以利下節進行演算法及權重項目的評估。

二、建立最佳路線的系統模型

(一) 評估校園疏散的最佳演算法

我們以模擬器建立三張不同的複雜地圖，透過不同的演算法進行觸及格數、運算成本和路徑格數的計算（圖 4-5），根據結果總結兩點如下：

1. 實際計算時還是以 Dijkstra 演算法雖然運算成本較高，但路徑格數最小。因此，我們依據 Dijkstra 演算法，使用 Python 語法設計校園疏散系統。
2. 我們嘗試以此模擬器建構出校園地圖，但發現此模擬器雖可以輔助我們了解演算法差異，卻由於介面設計的權重限制，使我們難以完整建構出校園地圖。

1號地圖				觸及格數	運算成本	路徑格數
dijkstra	平均代價					
	較大代價					
	較小代價					
	觸格格數在平均	154	597	1842	54	
	地圖平均代價					
	代價1					
a*	平均代價	167	295	767	54	
	較大代價	190	273	675	64	
	較小代價	154	586	1751	56	
	觸格格數在平均	154	506	1403	56	
	地圖平均代價	166	245	614	54	
	代價1	154	586	1751	56	
代價2	154	472	1319	54		

2號地圖				觸及格數	運算成本	路徑格數
dijkstra	平均代價					
	較大代價					
	較小代價					
	觸格格數在平均	81	603	1801	49	
	地圖平均代價					
	代價1					
a*	平均代價	89	162	284	53	
	較大代價	99	150	280	51	
	較小代價	99	150	280	51	
	觸格格數在平均	81	393	861	49	
	地圖平均代價	104	135	249	47	
	代價1	81	556	1507	49	
代價2	86	279	597	49		

3號地圖				觸及格數	運算成本	路徑格數
dijkstra	平均代價					
	較大代價					
	較小代價					
	觸格格數在平均	102.5	160	444	23	
	地圖平均代價					
	代價1					
a*	平均代價	122.5	102	225	29	
	較大代價	124.5	95	204	29	
	較小代價	102.5	122	315	23	
	觸格格數在平均	102.5	101	252	23	
	地圖平均代價	104.5	107	267	23	
	代價1	102.5	122	315	23	
代價2	104.5	116	295	23		

圖 4-5 以模擬器比較不同演算法的計算結果
(圖片由第二作者繪製)

(二) 權重的探討

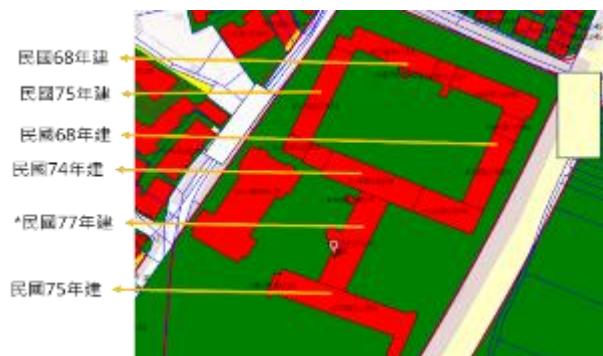
綜合研究一的資料調查與結果，影響疏散路線的各項因素，分別以自定義條件（表 4-6）、資料庫匯入（表 4-7）、實測與操作（表 4-8）等三項，詳細說明：

表 4-6 自定義條件的權重探討

權重項目	說明	備註
特殊生	學校特殊生的班級都安排在一樓，不列入考慮疏散難度。	不列入演算
出口規劃	規劃與建物的安全距離為最短逃生出口，將出口改為四個。 研究結果 1-3 校園地圖程式化中說明，p.14	地圖建模 定義
疏散順序	不同樓層間疏散時，以演算法的佇列邏輯，排定疏散排序： 1. 年級小優於年級大。 2. 同年級時，高樓層優於低樓層。 3. 同年級、同樓層時，依先抵達樓梯的位置先行。 4. 透過影像監控，當壅塞發生時可推薦班級改道 研究結果 3-2-openCV 中說明，p.11、p.19、p.24	演算法定義

研究的建物年齡中透過建物套繪地籍圖資料可知，所有建物年齡皆在民國 77 年前完工。詢問總務處得知，所有建物分別在 2016、2018 和 2021 年完成耐震補強設計。至截稿為止，無具體能評估不同建物間權重的差異，在這次研究中皆列為相同。

建物年齡



此次研究皆訂為相同

(圖片由第二作者繪製)

表 4-7 資料庫匯入的權重探討

權重項目	說明	備註
班級人數與位置	依據學校校務行政系統資料，將班級人數、上課地點、課表匯入資料庫。學生在班級教室、科任教室皆可取得最佳逃生路線的演算結果。 研究結果 3-1-使用 flutter 匯入說明，p.22-23	資料庫匯入

表 4-8 實際測量與操作的權重探討

權重項目	說明	備註
人流速度	根據不同年級學生進行行走速度的紀錄，並透過線性回歸的方式，賦予不同年級學生不同的權重值。 研究結果 2-3-人流速度計算，p.17-18	
走廊樓梯長寬	依據實測結果設計，研究因結果相近皆訂為相同。 研究結果 2-4，p.18	實驗定義 下小節詳述
壅塞示警	將樓梯口影像畫面透過 openCV 函數庫辨識人臉數量，當樓梯人臉數量大於 20 人、40 人時，調整該樓梯權重，並回傳訊息給 Arduino 進行壅塞示警。 研究結果 3-2-openCV 中說明，p.11、p.19、p.24	

(三) 實際測量與實作

1. 不同人數對於速度的影響

此實驗以中年級學生測試了 10 人、20 人、50 人時的速度差異，測試項目包含直線行走 45 公尺、行走 45 公尺距離轉彎一次、下樓的速度比較，由表 4-9 中可發現，人數越多時，確實對於速度造成影響。

我們將數據資料投入 Excel 試算線性回歸，取得人數與速度的方程式（如右圖）。透過圖 4-6 可知，直線與轉彎的人流速度差異極小，因此我們可以推估學生在走廊的直線與轉彎的人流速度訂為相同。其次，學生下樓梯的速度上，斜率有較大變化。10 人下樓速度最快，隨著人數增加，速度下降的越快。我們提取此次實驗的線性方程式（ $y=ax+b$ ），由表 4-10 說明數值代表的意義。

表 4-9 不同人數對速度的計算結果

	10人	20人	50人
直線	1.17	0.92	0.78
轉彎	1.19	0.98	0.72
下樓	5.4	3.2	2.76

單位:公尺/秒

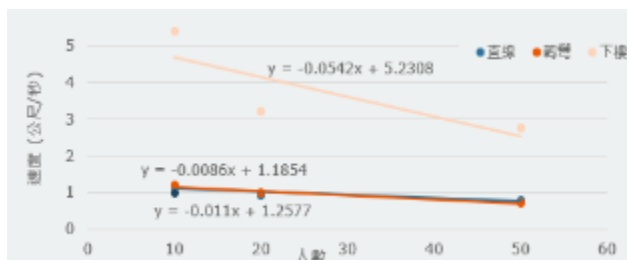


圖 4-6 人數與速度的方程式
(圖片由第一作者製作)

小，因此我們可以推估學生在走廊的直線與轉彎的人流速度訂為相同。其次，學生下樓梯的速度上，斜率有較大變化。10 人下樓速度最快，隨著人數增加，速度下降的越快。我們提取此次實驗的線性方程式（ $y=ax+b$ ），由表 4-10 說明數值代表的意義。

表 4-10 人數對於速度線性方程式的定義

	原始線性方程式	簡化方程式	人數(x)	速度(y)	斜率(a)	截距(b)
直線	$Y = -0.0086x + 1.1854$	$Y = -0.009x + 1.19$	x	y	-0.009	1.19
下樓	$Y = -0.0542x + 5.2308$	$Y = -0.054x + 5.23$	x	y	-0.054	5.23
方程式符號的意義			不同的人數	推測的速度	人數對速度的變化量	單人的速度基準值

2. 不同年級對於速度的影響

我們挑選低中高年級各二個班級（各 50 人）進行實驗，同樣測試包含直線行走 45 公尺、行走 45 公尺距離轉彎一次、下樓的速度比較，由表 4-9

表 4-11 不同年級對速度的計算結果

	高年級	中年級	低年級
直線	0.81	0.78	0.75
轉彎	0.79	0.72	0.73
下樓	4	2.76	1.09

單位:公尺/秒

中可發現，發現年級越高行走的速度明顯加快，但是在直線與轉彎的速度上三個年段差異並不大。而在下樓梯時的速度相較之下明顯較快；此外，低年級與中高年級在下樓的差異更是接近 0.5 公尺/秒，顯示樓層不同年級學生走樓梯有較大的速度差異。

因在校內無法要求大量的學生為我們的實驗進行實際測試，因此透過實驗結果 2-3-1 的資料，推估學生人數與速度關係的斜率（a 值）不變下，帶入低、高年級速度（y 值）和低、中、高年級測量人數（25 人）的條件，計算不同年級學生速度的截距，形成表 4-12 的低、中、高三個年段的不同人數與速度的方程式。

表 4-12 年級對於速度線性方程式的定義

環境	人數 (x)	速度 (y)	斜率 (a)	截距(求值) $b = y+ax$	形成不同年級學生的人數與速度方程式	
高年級	直線	25	0.81	-0.009	1.26	$Y = -0.009x + 1.26$
	下樓	25	4	-0.054	6.7	$Y = -0.054x + 6.7$
中年級	直線	25	0.79	-0.009	1.23	$Y = -0.009x + 1.23$
	下樓	25	1.09	-0.054	5.3	$Y = -0.054x + 5.3$
低年級	直線	25	0.75	-0.009	1.2	$Y = -0.009x + 1.2$
	下樓	25	1.09	-0.054	3.79	$Y = -0.054x + 3.79$

3. 綜合研究結果 2-3-1 和 2-3-2：

為了讓原本計算的「最短路徑」能進一步調整計算「最佳路徑」，「權重」調整為「1/速度」，表示速度越快的學生，獲得較低的權重，總結數據為表 4-13，以此權重作為不同年級學生經過不同走廊與樓梯的權重不同。

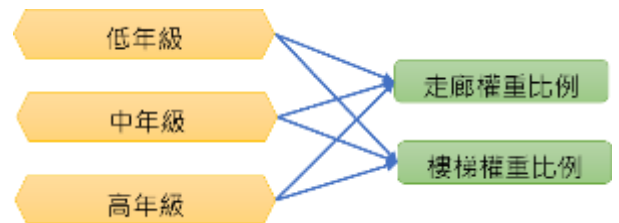


圖 4-7 人流速度的權重 (圖片由第二作者繪製)

表 4-13 不同年級學生在 25 人基準下的人流速度權重比例

25 人基準	低年級		中年級		高年級	
	速度	速度變數	速度	速度變數	速度	速度變數
直線	0.98	1.02	0.97	1.02	1.04	0.97
下樓	2.44	0.41	3.88	0.26	5.35	0.19

4. 校園走廊、樓梯長寬的測量

我們實際量測繪製校舍地圖（圖 4-7），總結權重如下：

- 校園教室寬度落於 8.7 公尺到 9.51 公尺間，而

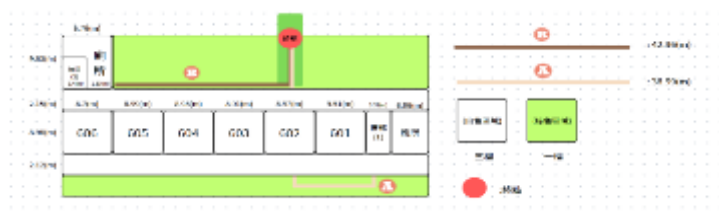


圖 4-8 測量校園走廊、樓梯長寬 (圖片由第二作者繪製)

每個學生的平均快走步伐約 0.78 公尺/步，表示教室間的距離誤差約為學生步伐一兩步時間，因此直接統一以 9 公尺計算每間教室距離。

- 量測多個樓梯寬度為 3.4 到 3.5 公尺，逃生時多半是以二路縱隊為規劃，此寬度大小也可取大概均值計算，列為相同。

(四) 實測系統

1. 程式模型

我們將最新的程式模型訂為二個部分：地圖建模（研究結果 1-3 說明，p.14-p.15）、演算法下的權重（研究結果 2-1 至 2-3 說明，p.15-p.18）等。進行程式模型的建構說明：

(1) 人流速度的權重

不同樓梯、走廊的權重根據低、中、高年級學生的速度不同（研究結果 2-3），產生不同的權重。計算不同年級在走廊與樓梯的不同速率比(以 25 人為基準)。

```
weight_adjustment = {
('Y1', 'room-edge'): 1.02,
('Y2', 'room-edge'): 1.02,
('Y3', 'room-edge'): 0.97,
('Y1', 'staircase-edge'): 0.41,
('Y2', 'staircase-edge'): 0.26,
('Y3', 'staircase-edge'): 0.19 }
```

圖 4-9 人流速度的權重程式碼
(圖片由第二作者繪製)

(2) OpenCV 的回饋權重

使用「可變」的權重計算，偵測 openCV 的人數改變樓梯權重。（整合至研究結果 3-2 說明）

```
#偵測 openCV 的人數改變樓梯權重
def adjust_stair_weight(person_count):
    if person_count > 40:
        return stair_node_weight =
G.nodes[path[0]]['node_weight']*3
    elif person_count > 20:
        return stair_node_weight =
G.nodes[path[0]]['node_weight']*2
    else:
        return stair_node_weight =
G.nodes[path[0]]['node_weight']*1
```

圖 4-10 OpenCV 的回饋權重程式碼
(圖片由第二作者繪製)

2. 演算法下的總權重

- A. 人流速度的權重：
- B. 路徑上每個節點的權重（代表路徑的壅塞度）：
使用起始節點的權重和當前邊的類型組合來調整權重
- C. 具有樓梯監視器的 OpenCV 回饋權重。
總權重 = 1 + 2 + 3。

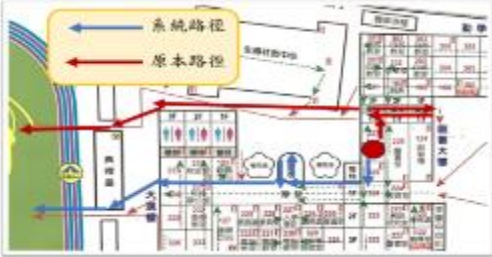
```
def calculate_path_weight(G, path, weight_adjustment):
    total_weight = 0
    # 獲取起始節點的權重
    start_node_weight = G.nodes[path[0]]['node_weight']
    for i in range(len(path) - 1):
        u, v = path[i], path[i+1]
        edge_data = G[u][v]
        edge_type = edge_data['type']
        # 計算人流速度的權重
        adjustment_factor =
        weight_adjustment.get((G.nodes[u]['node_type'], edge_type),
        1)
        adjusted_weight = start_node_weight * adjustment_factor
        total_weight += adjusted_weight
    # 加上路徑上每個節點的權重
    for node in path:
        node_weight = G.nodes[node]['node_weight']
        total_weight += node_weight
    # 加上 OpenCV 的回饋權重
    stair_node_weight = G.nodes[path[0]]['node_weight']
    total_weight += stair_node_weight
    return total_weight
```

圖 4-11 演算法下的總權重程式碼
(圖片由第二作者繪製)

3. 比較程式模型與實際結果

進行許多調查與設計後的程式模型，我們與目前現有的逃生路線進行速度的比較，其結果說明如下：

表 4-13 不同年級學生在 25 人基準下的人流速度權重比例

R502 路線比較	
R502 原本逃生路線	R502 演算出路線
R502 -> R501 -> E3 -> E2 -> E1 -> R912 -> R1012 -> Exit,	從 R502 到 Exit 的路徑: R502 -> R503 -> R504 -> R607 -> R431 -> R231 -> C3 -> C2 -> C1 -> R812 -> Exit, 路徑權重總和: 374.68
	
比較	
R502 位於二樓，原本路線是直接下樓梯到一樓再行走穿堂抵達出口；但是系統計算則是行走二樓至 C 樓梯再到一樓至出口，計算出權重則是原本路徑 > 系統路徑。	

(1) 系統執行計算與比較

系統實際的部分計算結果如上表，將 **Exit** 設定為全校逃生演練的避難點進行計算，可以得知所有節點（教室）的最佳路徑必須如何行走。因此透過執行成果與校

園目前的逃生地圖進行比對，發現幾個與原本路徑不同的結果，我們選擇 R502 路徑（表 4-13）說明：

(2) 程式模型驗證實測

得知系統的計算結果後，我們實際行走「原本路線」與「系統路徑」紀錄時間進行比較，實驗時為二人模擬二路縱隊方式行走兩個班級的路線計算時間，而行走過程中受試的人不得先行得知自己行走路徑的時間，實驗結果與實測照片如圖 4-12，並將行走時測量時間的結果列為表 4-14。



圖 4-12 實測行走示意圖
(照片由第二指導老師拍攝)

表 4-14 R502 行走路線實測結果

教室位置:R502						
路線	原本路線			系統路線		
次數	1	2	3	1	2	3
時間	2'57	2'55	2'51	2'09	2'10	2'09
結果	系統路線較快抵達疏散位置					

(五) 小結

1. 由實測結果顯示 R502 系統路線比原路線快，我們認為「原本路線會經過更多班級，推測疏散時會造成壅塞。」可能導致更大時間差異。
2. 這次的設計是以單一出口來進行模擬，但是與老師討論後，我們期待實際逃生會以最快離開建築物為目標設計，因此後續設計疏散系統時會更改為四個出口進行路線規劃。
3. 根據我們的系統模型顯示這個研究是具有可行性，進一步開始建立跨平台、可移動式的校園智慧疏散系統。

三、建立跨平台、可移動式的校園智慧疏散系統

製作完整的校園疏散系統時，我們思考以班級教室為出發點，我們將整個系統分為移動裝置（平板、手機）、固定裝置（教室內的大屏、計算路徑的主機與校內的監視器、蜂鳴器）兩個部分設計（圖 4-13），下列分項說明結果。

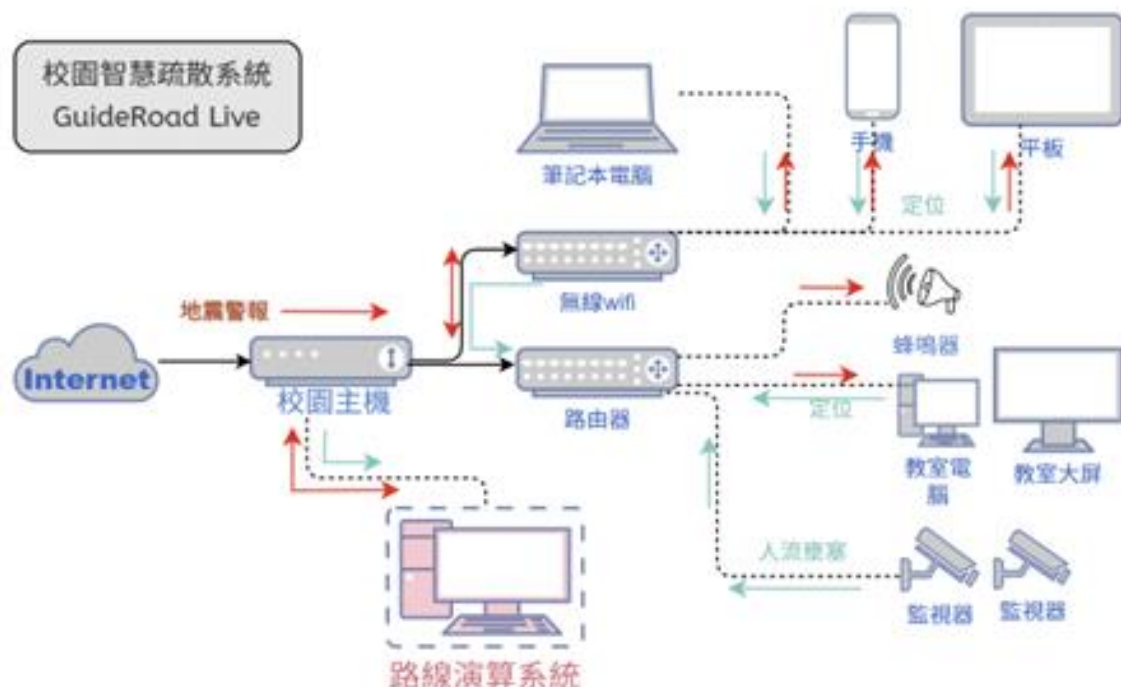


圖 4-13 跨平台、可移動式的校園智慧疏散系統概念圖
(圖片由第二作者繪製)

(一) Flutter 技術架構 (軟體)

下圖 4-14 是完成的 APP 介面說明，Flutter 可同時轉為網頁版本，介面功能相同，因此我們主要以 APP 來說明功能設計。

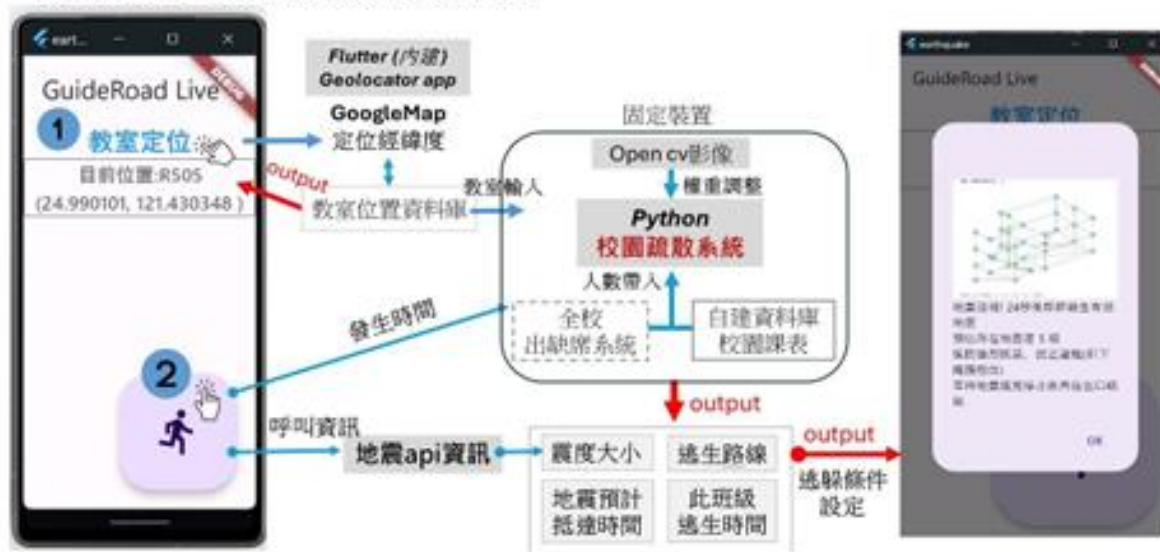


圖 4-14 Flutter 技術架構下的功能總覽
(圖片由第二作者繪製)

介面的設計主要有兩個按鍵**定位教室**與**模擬地震**按鍵，以及延後發生的**避難追蹤**。

1. 教室定位：

點下按鍵之後就能透過 **geolocator** 幫忙抓到經緯度（圖 4-15），並連結到我們是先建置完成的教室位置資料庫，確認出所在的位置在哪裡。

(1) 我們透過學校平板以 **google** 地圖抓取教室的經緯度（同一間教室範圍內走動抓取十次的經緯度資料），設定出一間教室大概的經緯度位置，例如箭頭教室位置 **R702** 經緯度



圖 4-15 geolocator 的教室定位
(照片由第一作者拍攝)

如表 4-15。為了降低誤差，這些資料同時透過 **Google map** 的經緯度定位再次確認。

表 4-15 R702 教室經緯度

	R606	R702	R607	R505	R407
經度範圍	24.989498-24.989593	24.99048-24.990544	24.989958- 24.990063	24.990076-24.990159	24.990088-24.990152
緯度範圍	121.4302-121.430274	121.4298-121.42996	121.430077-121.430171	121.430279-121.430391	121.430543-121.430648
	24.989519 121.430268	24.990517 121.4298161	24.990055, 121.430146	24.990087, 121.430279	24.990114, 121.430560
	24.989498 121.430274	24.9905167 121.4298403	24.990063, 121.430110	24.990159, 121.430299	24.990088, 121.430615
原始資料	24.989504 121.430200	24.9905495 121.4298014	24.989958, 121.430171	24.990121, 121.430391	24.990152, 121.430566
	24.989593 121.430246	24.9905535 121.429804	24.989984, 121.430077	24.990107, 121.430315	24.990122, 121.430648

(2) 抓到教室位置的經緯度後再抓取資料庫 **data**，定位出教室位置。部分程式碼如圖 4-16。

```
if (_currentPosition != null) Text(
    "LAT: ${_currentPosition.latitude}, LNG: ${_currentPosition.longitude}"
), // Text
FlatButton(

Future<Position?> getPositionById(String classId) async {
    final db = await database;
    final List<Map<String, dynamic>> maps = await db.query(
        'class_position',
        where: 'class_id = ?',
        whereArgs: [classId],
    );
```

圖 4-16 geolocator 教室定位程式碼
(圖片由第一作者拍攝)

(3) 圖 4-17 為實際操作介面的螢幕截圖與我們實際在校園內以手機測試操作的畫面。系統可成功抓到目前的位置，並顯示出定位出的經緯度。

- (2) 實際操作於校內的大屏畫面如下圖 4-20，模擬實際地震發生時，如果學生處於上課中狀態，便能馬上早於地震發生時，有更多預留的準備時間做出反應。



圖 4-20 校園大坪畫面的模擬測試
(照片由第二作者拍攝)

(二) openCV 的影像偵測與壅塞回饋



圖 4-21 使用 openCV 進行學生人數偵測，並將數值回傳演算法中樓梯的權重
(圖片由第一作者截圖與經學務處授權後翻拍自校內監視器)



圖 4-22 實際將人數偵測數值使用 esp32 呼叫，使蜂鳴器作動
(照片由第一作者拍攝與截圖)

(三) 避難追蹤的求援與情感機制

當較大地震發生時，很可能發生人員受困情形，因此在發出地震預報後設定為 10 分鐘，規劃程式再進行後續的避難追蹤，根據條件不同會帶出圖 4-21 的回饋訊息：



圖 4-17 實測 geolocator 抓取教室定位
(照片由第一作者拍攝)

2. 模擬地震

設計直接帶入地震 API 的 data，接著同時利用研究二所開發出的 Python 演算程式運算出最佳路徑，接著比對兩方資料後給予逃生或是避難的建議。

目前判斷機制是計算出的時間短於地震預測到達時間時，此外還要少於 15 秒鐘，這個時間差是由我們實際實驗計算學生從教室全部離開的時間，以下是我們最後計算的語法（圖 4-18）。

```
//地震模擬通知提醒語法
String nowPos = classPos.classId;
String exitTo = classPos.exitTo;
int needTime = int.parse(classPos.needTime.toString());

int intensity = randomGen(1, 7);
int sec = randomGen(5, 40);
_classPosController.text =
"地震預報！$sec秒後即將發生有感地震！\n" "預報所在地震度 $intensity 度！\n";
if (intensity > 3) {
int timeDifference = sec - needTime;
if (timeDifference < 15) {
_classPosController.text +=
"請仍保持鎮定，並迅速離開($exitTo)！\n" "等待地震搖晃停止後再往出口疏散！";
} else if (timeDifference > 15) {
_classPosController.text +=
"趕緊疏散！！\n" "(您所在位置[$nowPos] - 避難地的逃生時間[$exitTo] - 預知數$needTime)";
} else {
_classPosController.text += "就地避護！";
}
} else {
_classPosController.text += "請原地避護！";
}
}
```

圖 4-18 帶入地震 API 的 data 的地震計算
(圖片由第一作者截圖)

- (1) 圖 4-19 是操作 app 時實際獲得的通知畫面，系統根據輸入的地震震度與秒數，最後依據不同條件跳出不同的提醒警訊，說明是否要逃生，或是就地避難的提示說明。



圖 4-19 app 的通知畫面
(圖片由第一作者截圖)

- ◆ 有網路狀態且可以回覆的情況下，呼叫出 google 助理功能協助對外求援。有網路狀態但是沒辦法確定人員是否安全，便設定等待 120 秒後，自動對外求援。



圖 4-21 避難追蹤 APP 顯示畫面（有網路）

- ◆ 無網路狀態，自動連結關懷資料庫（表 4-16），程式帶入 google 語音轉文字功能抓取關鍵字，根據關鍵字回覆對應的語音訊息，下表為我們目前設定的關懷資料庫部分內容。

表 4-16 關懷資料庫

問題關鍵字	關懷回覆(記錄Chat GPT提供不同回答)
冷	讓我們試試一個放鬆技巧，好嗎？從你的腳開始，慢慢地緊攏然後放鬆每一塊肌肉，一直到頭頂
	感到害怕和冷是很正常的反應。這是你的身體在應對壓力。我會一直在這裡支持你。
	閉上眼睛，想像一個你感到安全和舒適的地方。試著感受那裡的溫暖和平靜。
痛	如果可以的話，試著輕輕地活動一下手腳，這樣能幫助你保持溫暖。
	如果可以的話，試著保持冷靜和放鬆，深呼吸有助於減少疼痛感，如果有冰袋或冷水，可以用來冰敷疼痛部位以減輕不適。
害怕	請注意你周圍的環境，確保自己處於安全的地方，如果需要移動，請慢慢來，確保不會再受傷。
	感到害怕是很正常的，特別是在這種情況下。我理解你的感受
	我會一直在這裡陪著你，你並不孤單。我們可以一起度過這段時間。
	讓我們一起做幾次深呼吸，深深吸氣，然後慢慢呼氣，這樣有助於放鬆。
	閉上眼睛，想像一個你感到安全和舒適的地方，也許是一個你喜歡的公園或家裡的某個角落

（四）小結

透過 Flutter 技術架構建立的校園智慧疏散系統，綜合了研究一、二中的地圖建模和演算法系統模型，更進一步加入了帶入校園班級資料、個人化定位、呼叫地震資料、即時回饋和避難追蹤等功能，使系統模型得到實踐的機會，並且在校園中已讓部分學生開始體驗，讓我們的研究讓更多人看見。

伍、結論

一、建立最佳疏散系統模型：

為了建立系統我們先行進行了多項資料調查與分析，包含訪問調查、錄影畫面分析、演算法認識，這讓我們聚焦了疏散過程中需要考慮的多種因素，包含：

- (一) 將校園地圖簡化成立體節點圖，再由模擬器比較分析確認了適合系統的演算法，最後利用 Dijkstra 演算法結合 Python 程式設計，進行了疏散系統的設計。
- (二) 根據研究結果二、三地逐步說明演算法下的權重，討論人流速度、走廊樓梯長寬、班級特殊生、班級人數與位置、出口規劃、疏散順序、壅塞示警、建物年齡等項目，盡可能的探究各種影響疏散結果的因素。

二、建立跨平台、可移動式的校園智慧疏散系統：

隨著系統模型建立，實際測試演算結果，最終建構了一個校園智慧疏散系統，結合了精準定位、壅塞回饋、避難追蹤等功能，達到聰明疏散的目標，為維護校園們師生的安全提供一個更全面性的建議，系統發展架構如下圖 5-1。

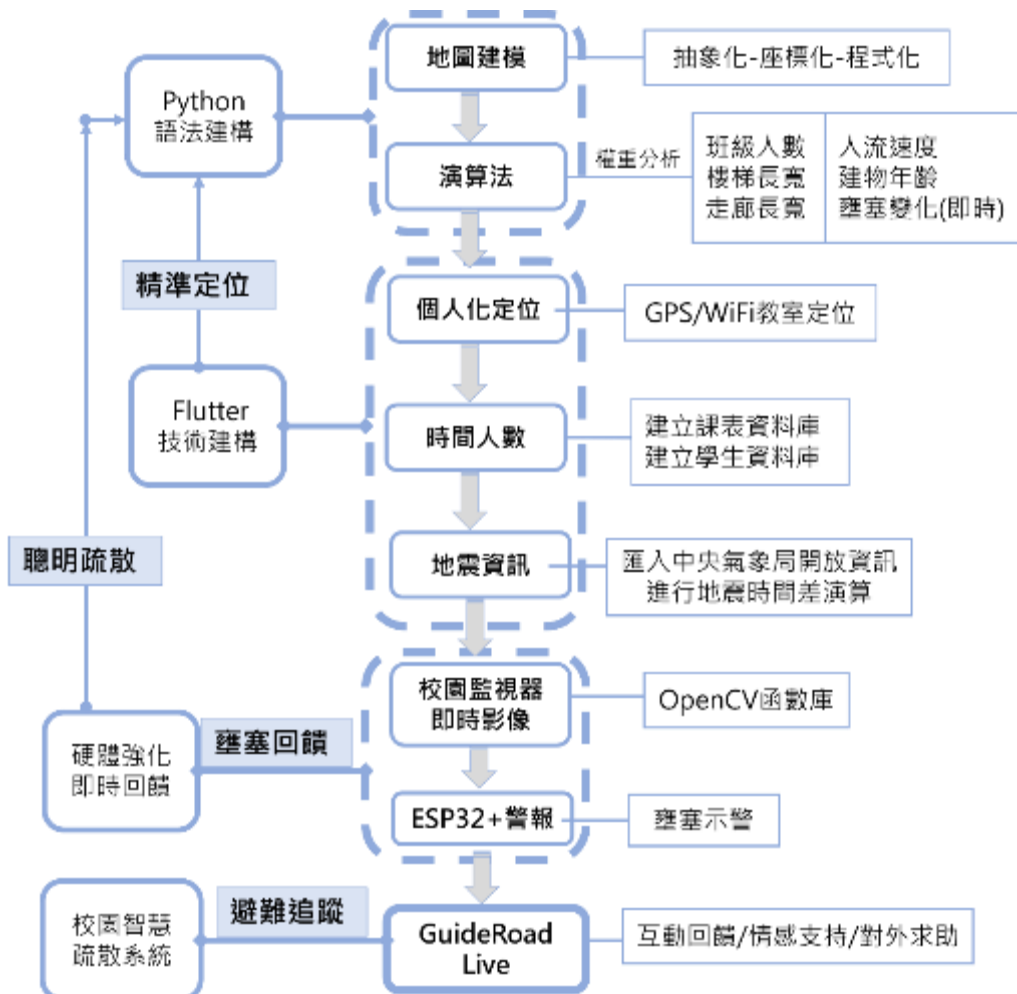


圖 5-1 跨平台、可移動式的校園智慧疏散系統
(圖片由第一作者與第二作者共同繪製)

陸、參考文獻

1. 吳東城, 卓祥琳, & 王晏慈. (2012). 一線生機－智慧型災難逃生引導系統。全國中小學科展作品第 52 屆作品。
2. 黃進興, & 陳建忠. (2012). 學童避難速度調查研究。建築學報, 80(S), 71-87.
<https://doi.org/10.6377/JA.201206.0012>
3. 謝馨瑤. (2013). 校園逃生出口空間疏散避難分析之研究—以校園室內空間為例（碩士論文，中華科技大學）。臺灣博碩士論文知識加值系統。<https://hdl.handle.net/11296/9erjs7>
4. Recording everything. (n.d.). 基礎演算法系列 — Graph 資料結構與 Dijkstra' s Algorithm。Medium。 <https://medium.com/技術筆記/基礎演算法系列-graph-資料結構與-dijkstras-algorithm-6134f62c1fc2>
5. 沈宸葳. (2017). 智慧型室內防災疏散系統之設計與規劃（碩士論文，東方設計學院）。臺灣博碩士論文知識加值系統。<https://hdl.handle.net/11296/462243>
6. 渡部有隆, & Mirenkov, N. (2022). 會動的演算法：61 個演算法動畫 + 全圖解逐步拆解，人工智慧、資料分析必備。旗標出版社。
7. Steam 教育學習網. Python Numpy 教學 - Array Create。 <https://steam.education/python-numpy>
8. 路徑搜尋模擬器. <https://pathfindingdemo.gamelet.online/>
9. 中央氣象署氣象資料開放平台. <https://opendata.cwb.gov.tw/>
10. Flutter 官方網站範例. <https://flutter.dev.org.tw/>
11. 探索科技-中央氣象局 EEW 強震即時警報 (Earthquake Early Warning).
<https://github.com/ExpTechTW/TW-EEW>

【評語】 082813

1. 在這份作品中，學生從實際的校園安全需求出發，設計了一套智能化的校園疏散系統。
2. 研究方法使用人工智慧工具協助撰寫程式為一特色，並將校園圖資輸入系統中。在建立最佳疏散路線的系統模型，使用演算法，並納入權重的概念。
3. 此系統不僅考慮了正常疏散情況，還加入了地震情況下的特殊處理，如即時警報和求援機制，經過實際測試驗證了系統的有效性，並與現有疏散路線進行比較，是一個結合理論與實踐、技術與人文關懷的優秀作品。

作品簡報



百密一「疏」~

校園智慧疏散系統之建構

摘要

本研究建構校園智慧疏散系統—GuideRoad Live，協助學校規劃逃生路線，在緊急狀況下迅速提供疏散建議。該系統具備精準定位、壅塞回饋、避難追蹤、聰明疏散等四大功能(圖1)。首先透過地圖建模，結合校園WiFi和GPS實現個人定位。藉由校園監視器分析樓梯壅塞，即時更新數據、路線，與設計蜂鳴器警報。系統匯入地震資訊評估逃生時間，提供逃或躲的疏散意見，並對外援助與給予使用者關懷。最後，系統根據節點、邊、人流速度、壅塞回饋的即時數據等因素進行權重計算，達到聰明疏散的目標。

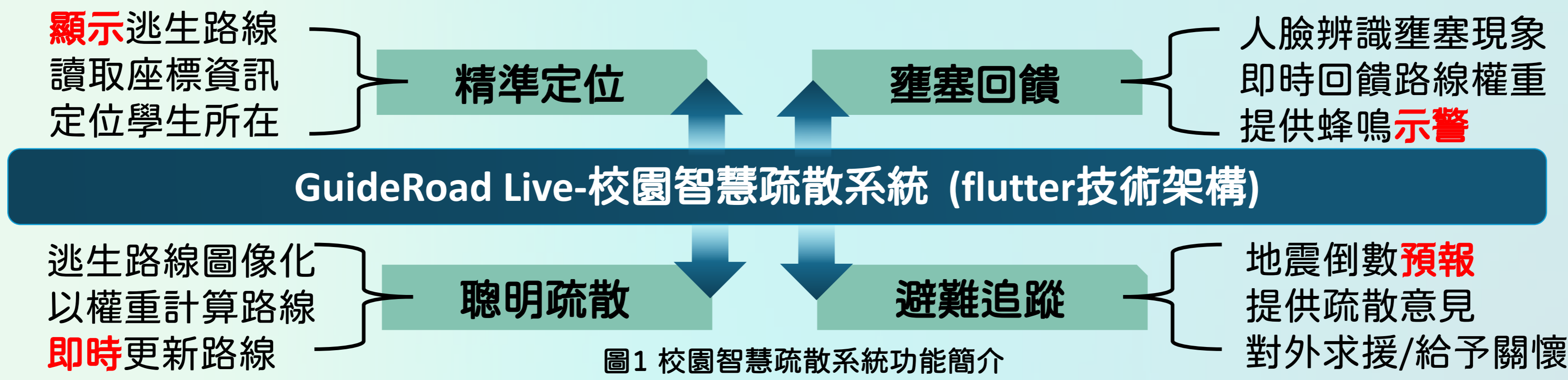


圖1 校園智慧疏散系統功能簡介 (圖片由第二作者繪製)

前言

一、研究動機

- (一)目前的疏散路線是否是最佳解呢？
- (二)壅塞發生時有沒有解決辦法呢？(圖2)
- (三)如果真的產生地震，該逃還是要躲呢？
- (四)遇到緊急危險時，有沒有人能幫助我呢？

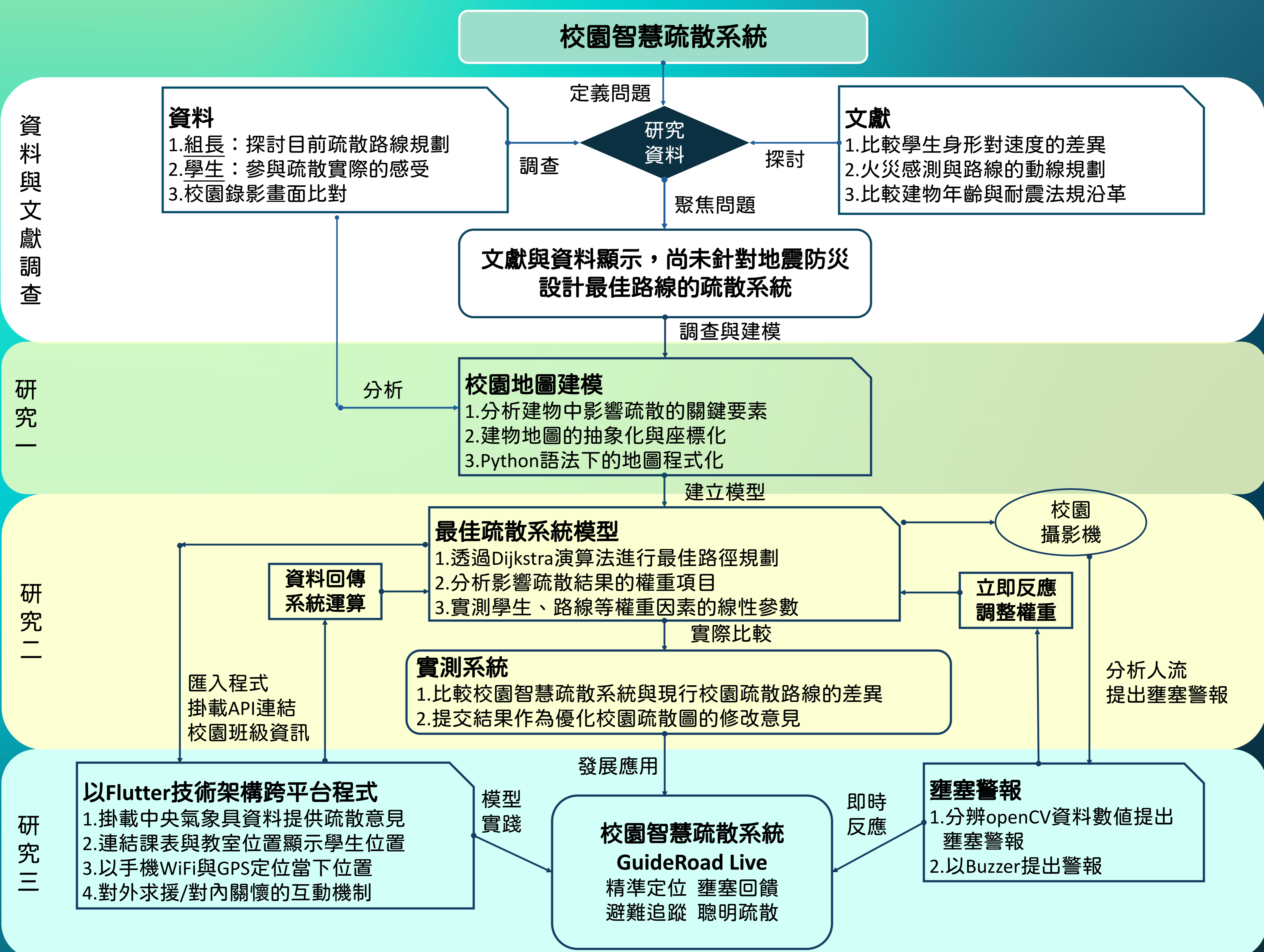
二、研究目的

- (一)校園地圖建模。
- (二)分析最佳疏散系統的模型。
- (三)建立跨平台，可移動式的校園智慧疏散系統



圖2 校園疏散時的壅塞現況 (照片由第一作者拍攝)

研究架構



資料與文獻調查

一、資料

(一)師長訪談與學生問卷調查(表1)

組長 (3名)	1.高年級在高樓層、低年級排在低樓層。 2.特殊生的班級，班級教室安排在一樓。 3.考慮到走廊寬度安排班級同時行走。
學生問卷 (10名)	1.出口大小與人數。

(二)校園錄影畫面比對(表2)



二、文獻

- (一)最佳路徑為考慮多因素下最快能夠抵達出口的路線
- (二)權重是Dijkstra演算法評估經過路線成本的數值。
- (三)校園疏散地圖(圖3)：
 - 1.疏散人流皆匯集前往操場
 - 2.部分班級穿越另一座建物
 - 3.無考慮年級間的速度差異
 - 4.無訂立壅塞情況下的反應

(四)逃生相關文獻

- 1.內政部築研究報告-學童避難速度調查研究。
- 2.沈宸葳(2017)-智慧型室內防災疏散系統之設計與規劃。
- 3.縣市工務局提供建物套繪地籍圖資料(圖3)。

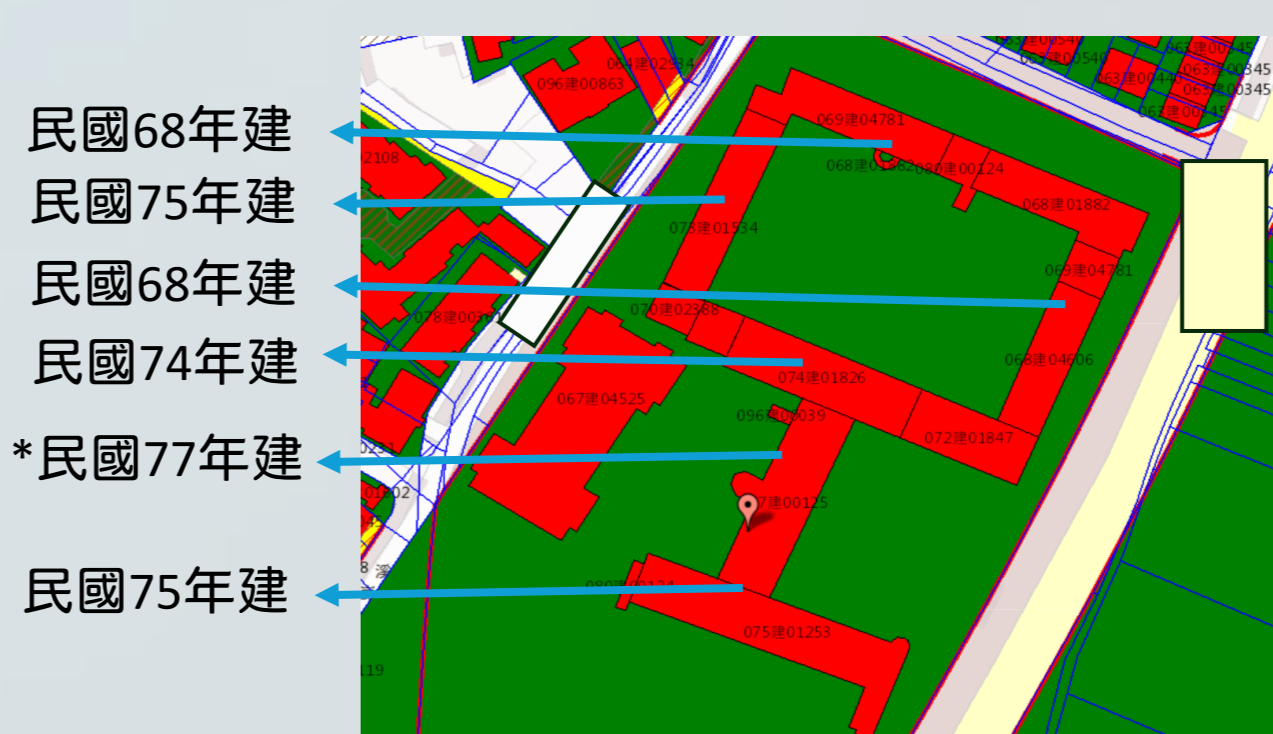


圖3 地籍圖資料 (圖片由第二作者繪製)

(五)影響校園疏散因素總表(表3)

	出口大小	出口位置	走廊長寬	年級	人數	樓梯大小	路線轉彎處
組長	●		●	●	●	●	
問卷	●				●		
文獻	●		●	●	●	●	
監視器	無差異	無差異	無差異	◆	◆	◆	◆
實作				◆	◆	◆	◆

研究結果一實作確認

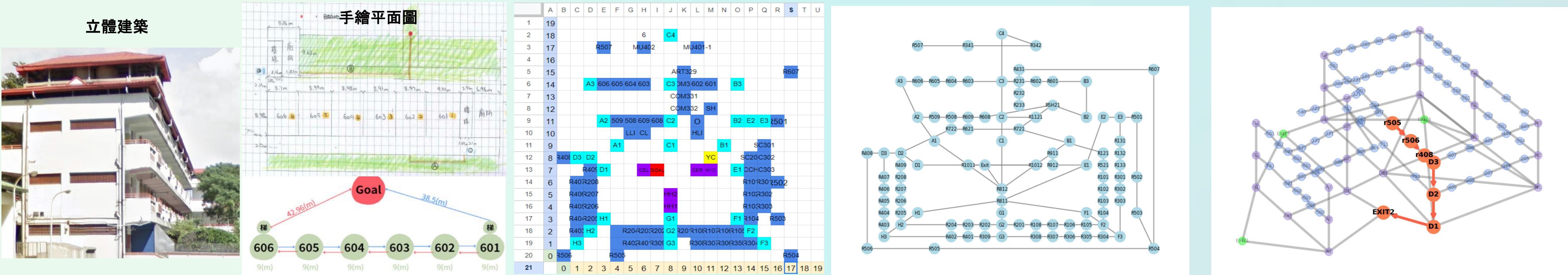
研究一、校園地圖建模

抽象化

座標化

程式化

立體化



- 小結：
1. 將出口、樓梯、教室等條件作為節點(node)進行建模，並利用networks圖形上邊(edge)的資料連結各節點。
 2. 以「立體圖形」方式呈現路線圖(圖4)，將逃生出口修正為4個，並在路線移動加上「箭頭」。

圖4 校園地圖建模歷程圖 (圖片由第二作者繪製)

研究二、分析最佳疏散系統的模型

一、評估校園疏散的最佳演算法

以模擬器建立三張不同的複雜地圖(表4)，根據權重表(圖5)決定以Python語法，使用Dijkstra演算法。

表4 模擬三種地圖比較演算法的差異(照片由第一作者拍攝)

比較	1號地圖				2號地圖				3號地圖			
	權重	觸及格數	運算成本	路徑格數	權重	觸及格數	運算成本	路徑格數	權重	觸及格數	運算成本	路徑格數
Dijkstra	154	597	1842	54	81	603	1801	49	102.5	160	444	23
A*	154	586	1751	56	81	556	1507	49	102.5	122	315	23

分為權重，觸及格數、運算成本和路徑格數等四個向度

比較	1號地圖				2號地圖				3號地圖			
	權重	觸及格數	運算成本	路徑格數	權重	觸及格數	運算成本	路徑格數	權重	觸及格數	運算成本	路徑格數
dijkstra	154	597	1842	54	81	603	1801	49	102.5	160	444	23
a*	154	586	1751	56	81	556	1507	49	102.5	122	315	23

圖5 模擬器計算地圖權重(照片由第一作者拍攝)

二、權重的探討

綜合資料與文獻調查，將影響疏散路線的因素，分為自定義條件(表4)、資料庫匯入(表5)、實測與操作(表6)說明之。

表4 自定義條件

權重項目	說明
特殊生	特殊生的班級都在一樓，不列入考慮。
出口規劃	4個最短逃生出口。
疏散順序	1.年級小 2.高樓層 3.先抵達 4.壅塞改道。
建物年齡	建物年齡相近，已完成耐震補強，無具體差異。

表5 資料庫匯入

權重項目	說明
班級人數與位置	班級人數、上課地點、課表匯入資料庫。

表6 實際測量與實作

權重項目	說明
人流速度	不同年級、不同路線下，賦予不同的權重值。
走廊樓梯長寬	根據實驗結果皆訂為相同。
壅塞示警	校園監視器分析樓梯擁擠程度(openCV語法邏輯)。

三、實際測量與實作

(一)不同人數對於速度的影響(表7)

以中年級學生測試了10人、20人、50人時的速度差異，將資料進行線性回歸(圖6)，取得人數與速度的方程式。

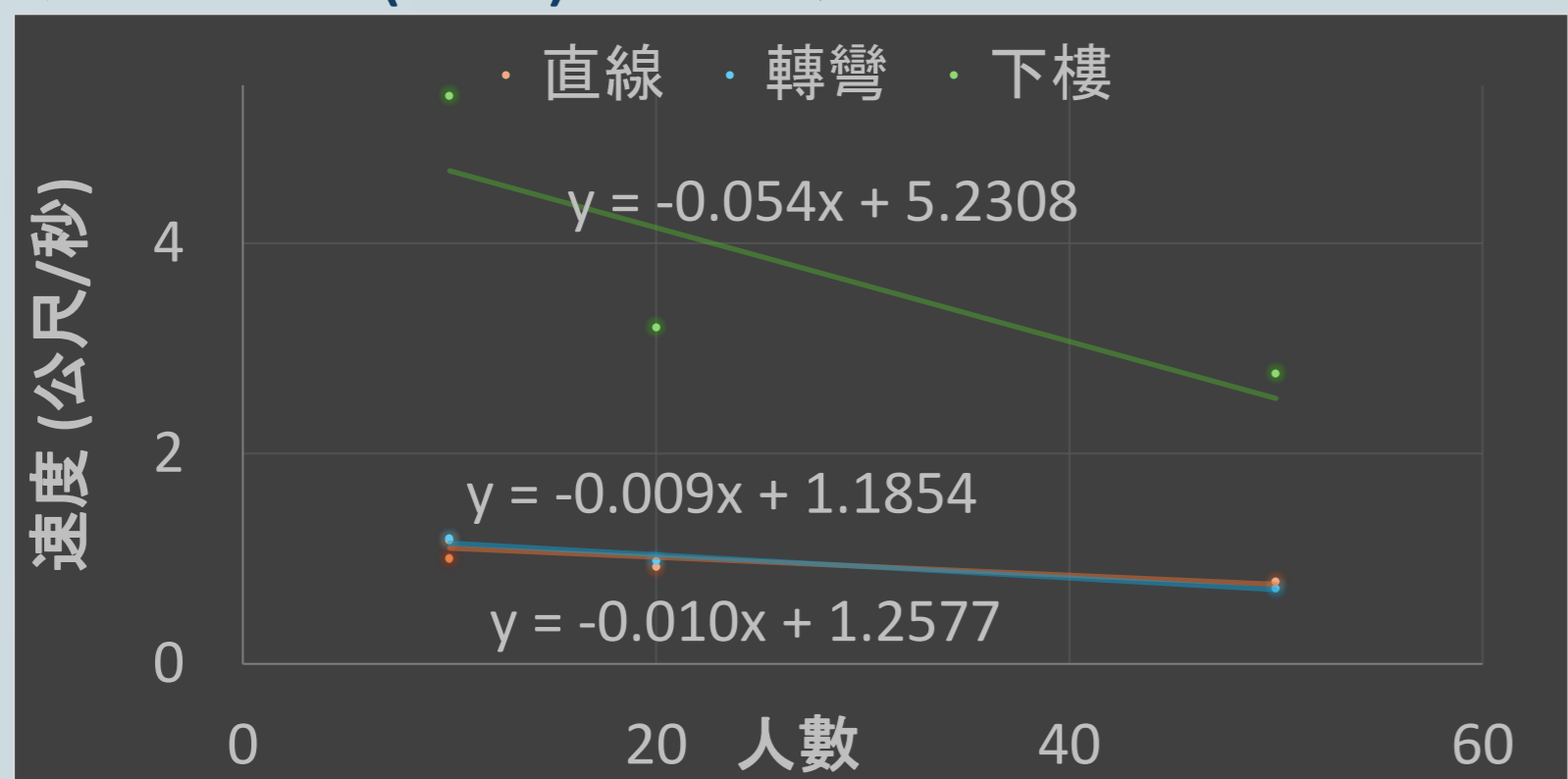


圖6 人數與速度的方程式(圖片由第二作者繪製)

表7 人數對速度方程式

不同人數	環境	人數	速度	斜率	截距(求值)	人數與速度方程式
不同	直線	x	y	-0.009	1.19	$Y = -0.009x + 1.19$
	下樓	x	y	-0.054	5.23	$Y = -0.054x + 5.23$

(二)不同年級對於速度的影響(表8)

我們挑選低中高年級各二個班級(各50人)的速度差異，將資料進行線性回歸，取得年級與速度的方程式。

表8 不同年級對速度方程式

	環境	人數	速度	斜率	截距(求值)	人數與速度方程式
高年級	直線	25	0.81	-0.009	1.26	$Y = -0.009x + 1.26$
	下樓	25	4	-0.054	6.7	$Y = -0.054x + 6.7$
中年級	直線	25	0.79	-0.009	1.23	$Y = -0.009x + 1.23$
	下樓	25	1.09	-0.054	5.3	$Y = -0.054x + 5.3$
低年級	直線	25	0.75	-0.009	1.2	$Y = -0.009x + 1.2$
	下樓	25	1.09	-0.054	3.79	$Y = -0.054x + 3.79$

「權重」為「1/速度」，表示速度越快的學生，獲得較低的權重。權重邏輯和程式碼為圖7，代表不同年級學生經過不同環境(樓梯或走廊)的權重不同。

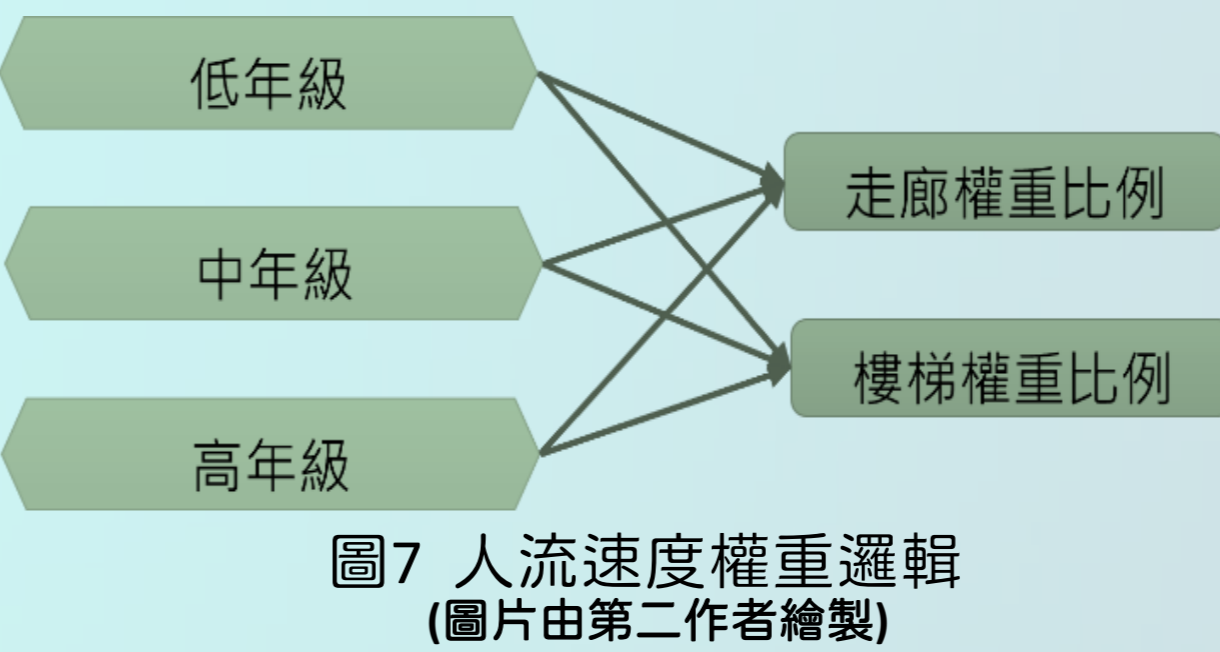


圖7 人流速度權重邏輯 (圖片由第二作者繪製)

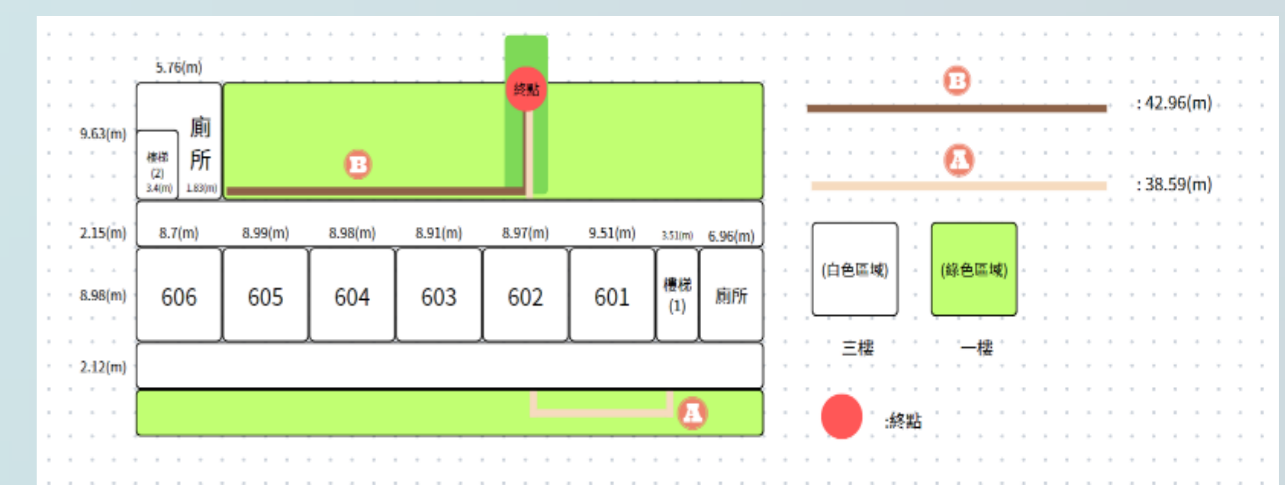


圖8 走廊寬度測量標準 (圖片由第二作者繪製)

(三)校園走廊、樓梯長寬的測量(圖8)

校園教室寬度落於8.7公尺到9.51公尺間；多個樓梯寬度為3.4到3.5公尺，考量學生行走的步幅落差，將本權重列為相同。

(四)演算法下的總權重：

1. 人流速度權重 = node權重(人數) * 環境權重比例。
2. 路徑上每個節點的權重(代表路徑的壅塞度)。
3. 校園監視器回傳的即時壅塞權重(即時修正環境現況)。
4. 總權重 = 1 + 2 + 3

四、實測系統

(一)R502的舊有與系統路線行進時間的比較(圖9)如下：



圖9 R502教室系統與舊有逃生路線比較表與時間測試 (圖片由第二作者繪製)

1. R502系統路線比原路線快，真實狀態下「原本路線會經過更多班級，推測疏散時會造成壅塞」可能導致更大時間差異。
2. 本次測試是以單一出口來進行模擬。
3. 根據我們的系統模型顯示本研究具有可行性，進一步開始建立跨平台、可移動式的校園智慧疏散系統。

(二)盤點系統與舊有路線的行走班級數(圖10)，統計如下：

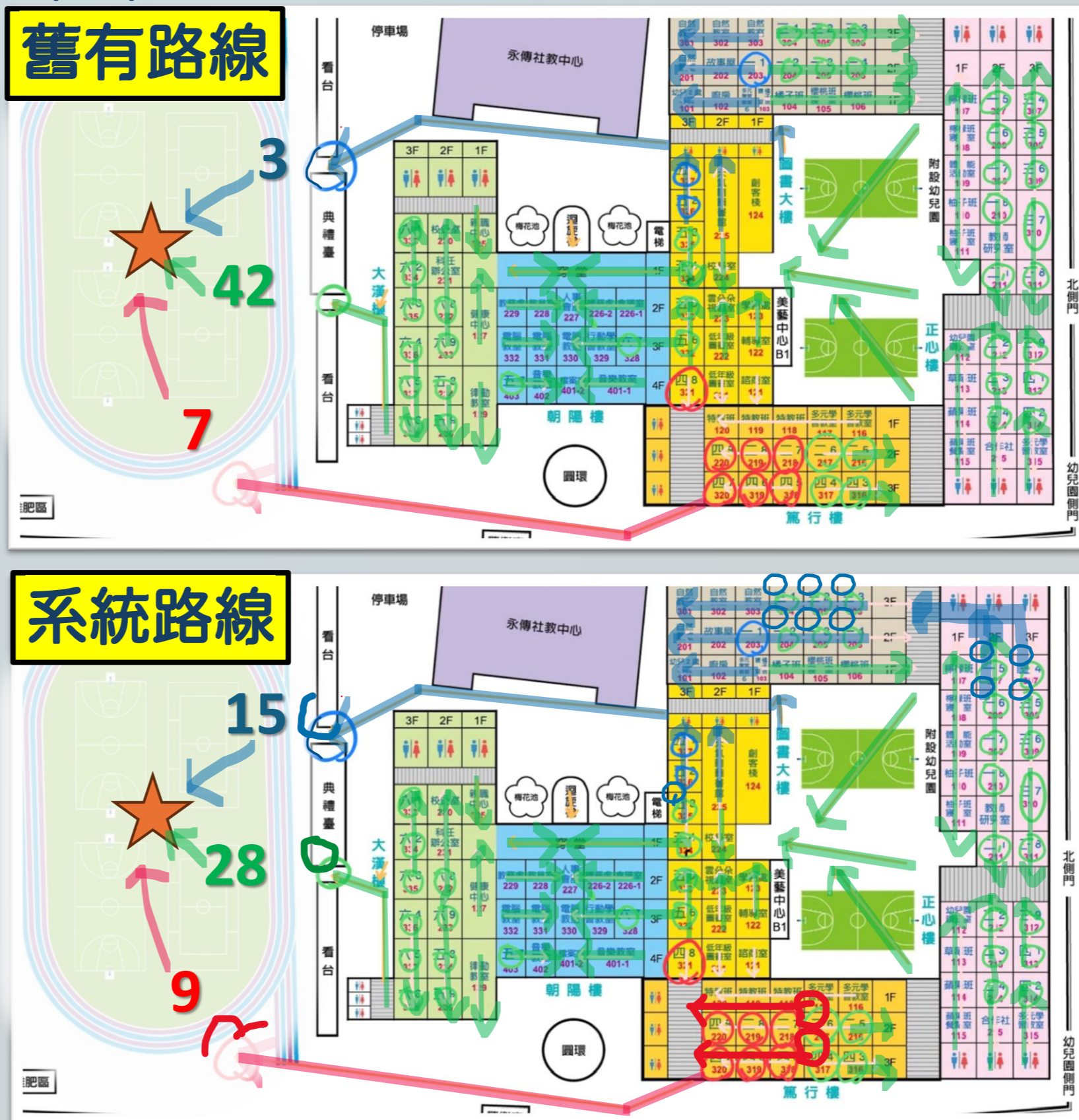


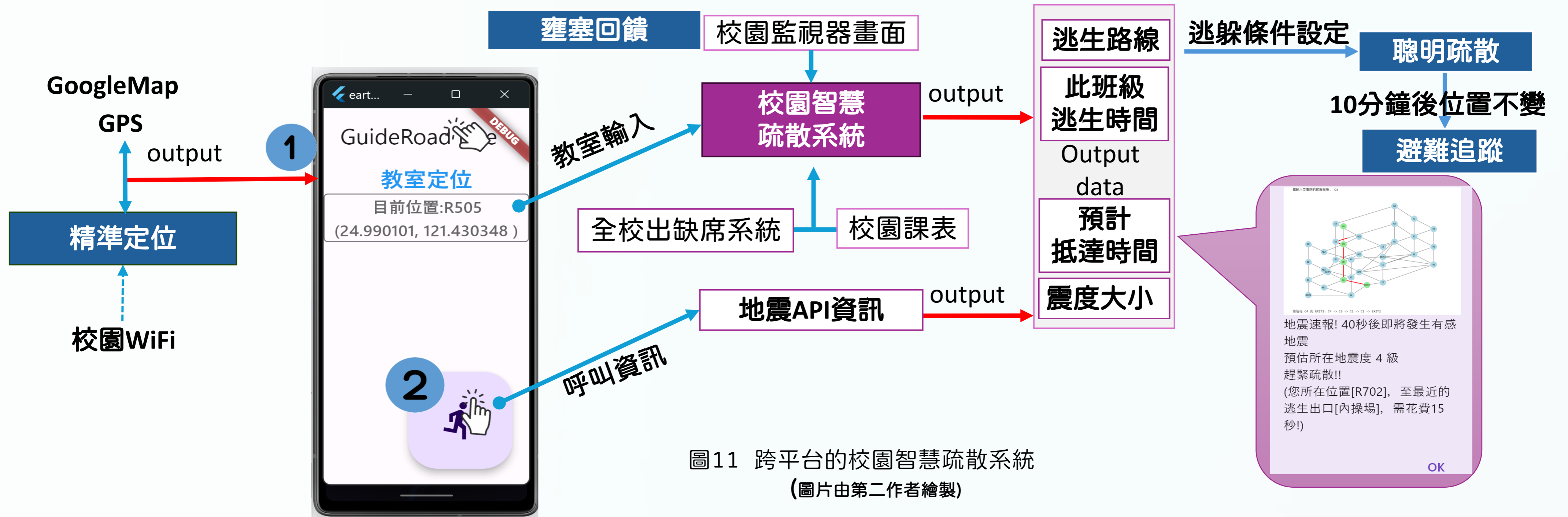
表9 統計不同路線行經班級數量

	系統路線	目前路線
路線(上)	15	3
路線(中)	28	42
路線(下)	9	7

根據系統路線與舊有路線的班級數量(表9)，可以看出系統路線有較好的分配性，減少壅塞的發生。

圖10 統計系統與舊有路線上的行經的班級數量(圖片由第二作者繪製)

研究三、建立跨平台、可移動式的校園智慧疏散系統(圖11)



一、精準定位

- 1.以手機或筆電連結教室智慧大屏的WiFi。
- 2.以geolocator幫忙抓到經緯度(GPS)(圖12)。
- 3.教室內走動測量10次，進行教室定位。
- 4.取得班級與教室課表，匯入資料庫加速演算。



圖12 geolocator抓到GPS (圖片由第二作者繪製)

二、壅塞回饋(圖13)

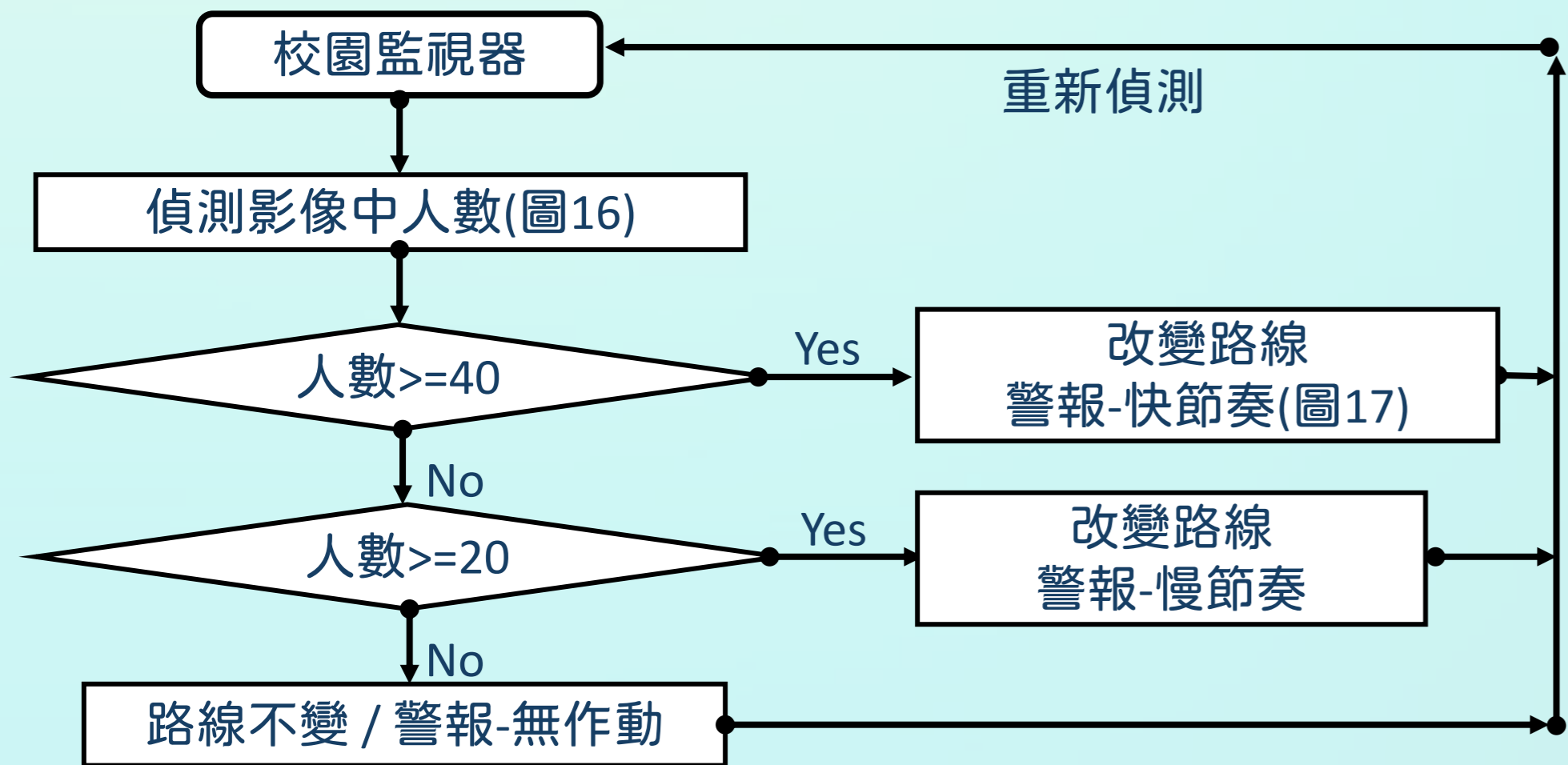


圖13 壅塞回饋邏輯圖(圖片由第二作者繪製)

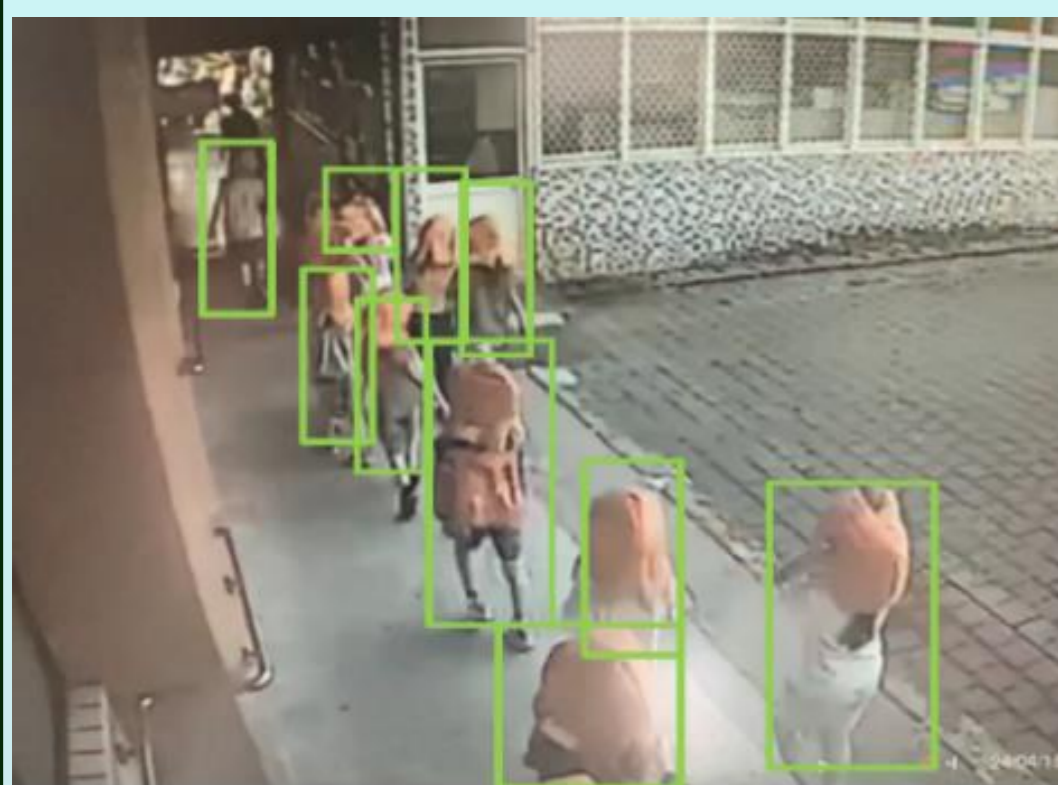


圖16 監視器分析人數 (照片由第一作者拍攝)



圖17 Arduino IDE的蜂鳴器作動 (照片由第一作者拍攝)



圖18 教室內模擬逃/躲警報 (照片由第一作者拍攝)

三、避難追蹤(圖14)

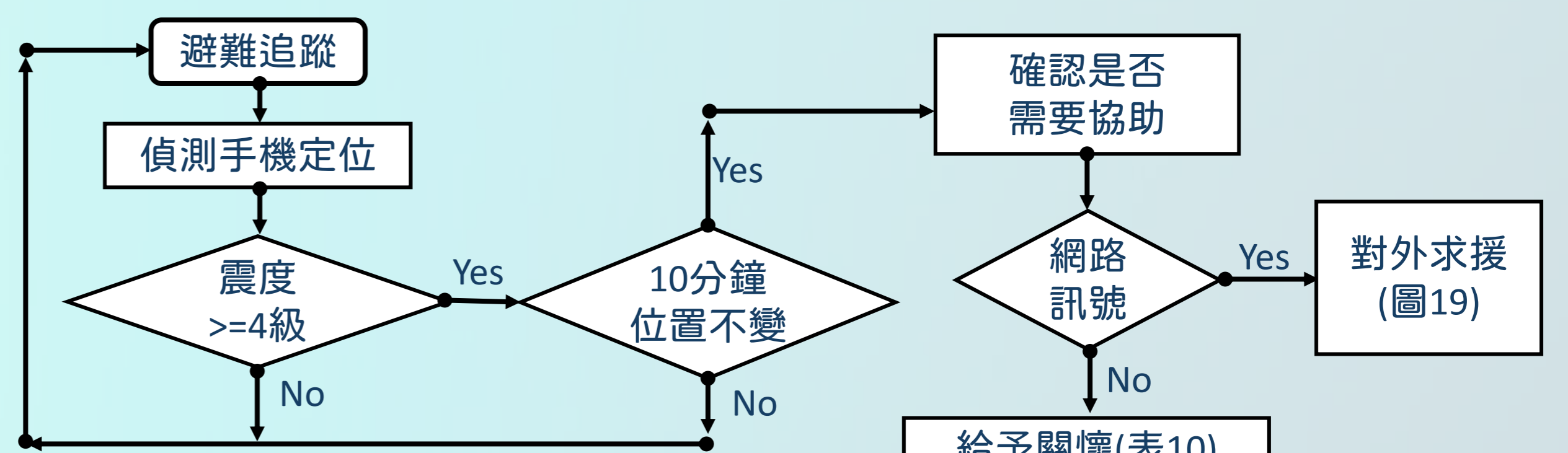


圖14 避難追蹤邏輯圖(圖片由第二作者繪製)

四、聰明疏散(圖15)

- 1.取得中央氣象局公開資訊的API與JSON資料檔。
- 2.取得「預估到達秒數」與「區域震度(PGA)」。
- 3.當【地震抵達時間】大於 預計逃跑時間+15秒→ **逃**
【地震抵達時間】小於 預計逃跑時間+15秒→ **躲**

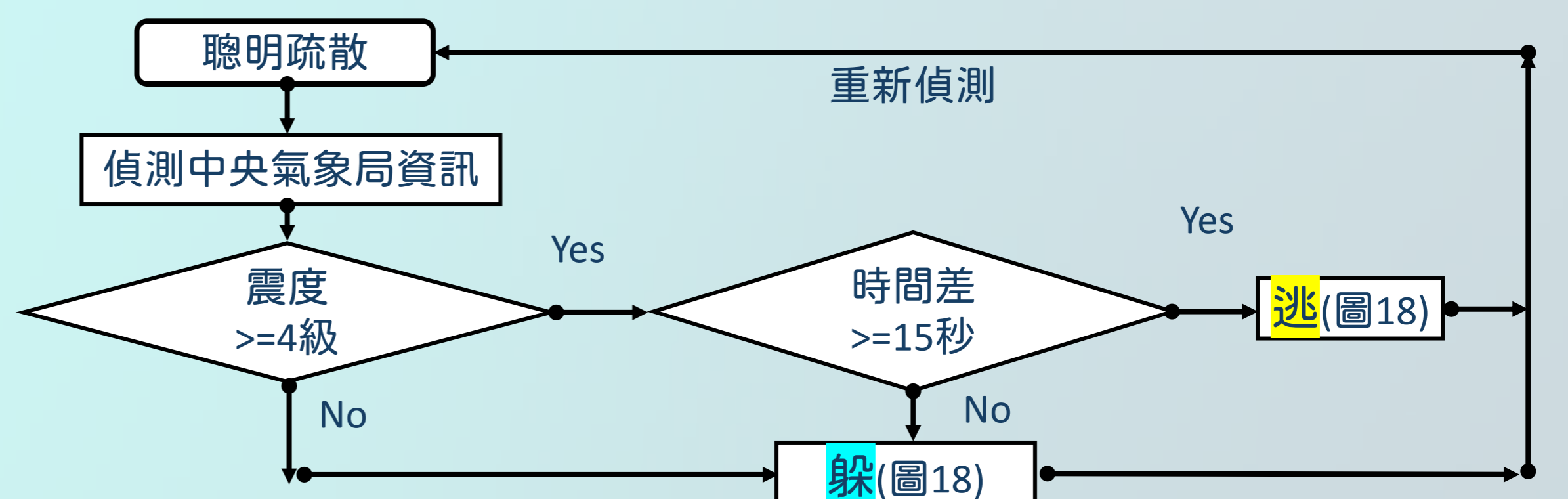


圖15 聰明疏散邏輯圖(圖片由第二作者繪製)

表10 無網路下的關懷資料庫

關懷	感受	冷
	感受	讓我們試試一個放鬆技巧，從你的腳開始，放鬆每一塊肌肉，一直到頭頂。感到害怕和冷是很正常的反應，這是你的身體在應對壓力。我會一直在這裡支持你。
	痛	如果可以的話，試著保持冷靜和放鬆，深呼吸有助於減少疼痛感。請注意你周圍的環境，確保自己處於安全的地方。如果需要移動，請慢慢來。
	扭傷	你可以用彈性繃帶包紮壓迫，可避免患處繼續腫脹，但切勿過緊，才不至造成末梢血液循環不良。
	急救	可抬高患肢高於心臟，以促進血液循環，避免患處腫脹。
	口渴	每次喝水量應該不超過幾口，以避免過快消耗。
		如果只剩少量水，建議每小時小口喝一次，以保持口腔濕潤並減少脫水的風險。



圖19 逃/躲通知及避難追蹤回報 (照片由第一作者拍攝)

結論

一、校園智慧疏散系統

結合精準定位、壅塞回饋、避難追蹤等功能，達到聰明疏散的目標(圖11)。

二、疏散系統模型

透過自建校園節點圖，參考並調整各項權重，結合校園監視器的影像可即時調整逃生路線。

三、未來展望

在未來的辦公大樓、各式商場建置智慧疏散系統(圖20)

- 1.大屏幕:投影大型的疏散地圖。
- 2.監視器:即時更新路線。
- 3.個人手機:互動回饋，安撫人心。

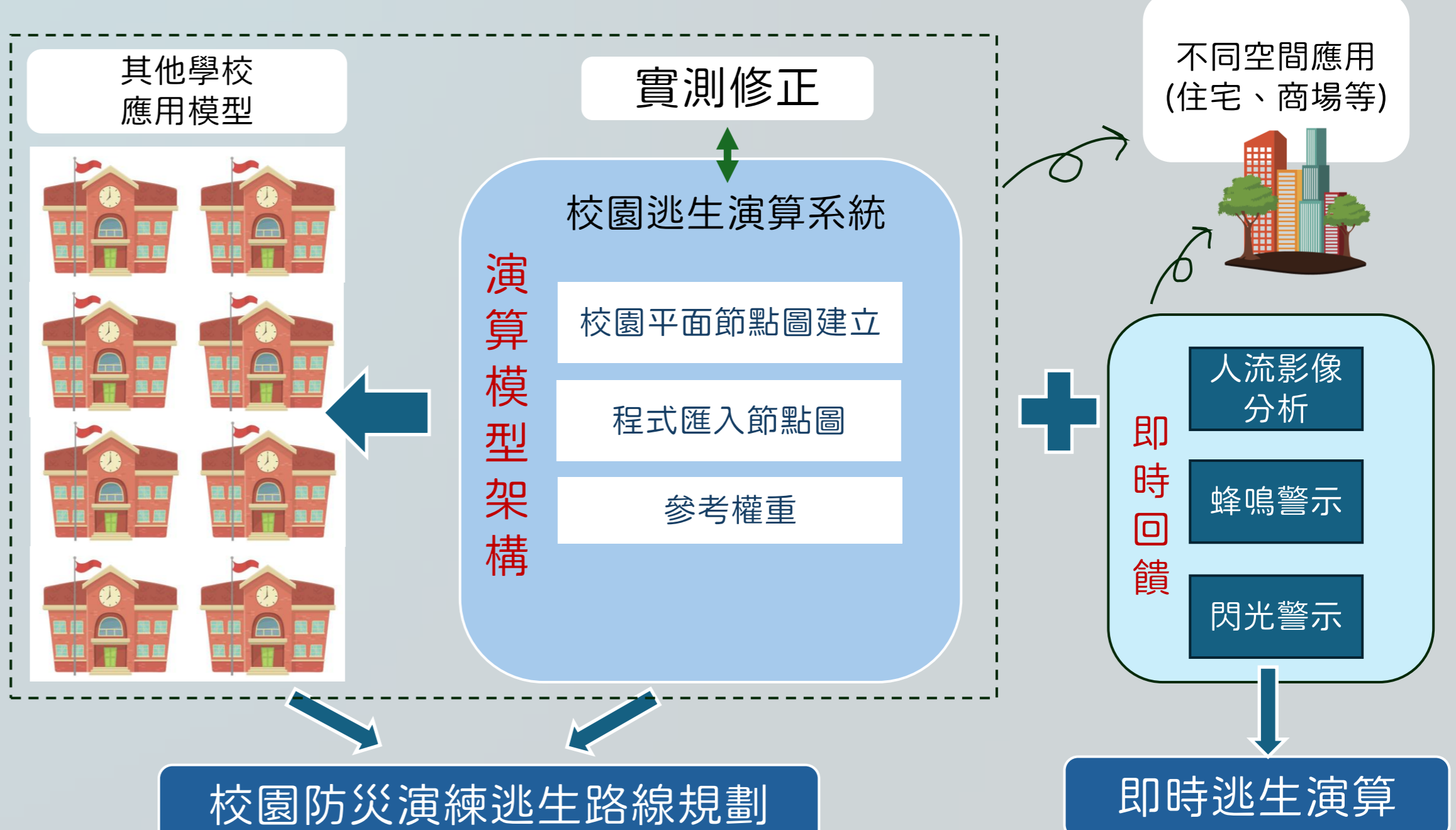


圖20 未來展望 (圖片由第二作者繪製)

參考文獻

1. 吳東城, 卓祥琳, & 王晏慈. (2012). 一線生機—智慧型災難逃生引導系統。全國中小學科展作品第52屆作品。
2. 黃進興, & 陳建忠. (2012). 學童避難速度調查研究。建築學報, 80(5), 71-87. <https://doi.org/10.6377/JA.201206.0012>
3. 路徑搜尋模擬器。 <https://pathfindingdemo.gamelet.online/>
4. 探索科技-中央氣象局 EEW 強震即時警報 (Earthquake Early Warning)。 <https://github.com/ExpTechTW/TW-EEW>
5. 謝馨瑤. (2013). 校園逃生出口空間疏散避難分析之研究—以校園室內空間為例 (碩士論文, 中華科技大學)。臺灣博碩士論文知識加值系統。 <https://hdl.handle.net/11296/9eris7>
6. Recording everything. (n.d.). 基礎演算法系列 - Graph 資料結構與 Dijkstra's Algorithm. Medium. <https://medium.com/技術筆記/基礎演算法系列-graph-資料結構與dijkstras-algorithm-6134f62c1fc2>
7. 沈震威. (2017). 智慧型室內防災疏散系統之設計與規劃 (碩士論文, 東方設計學院)。臺灣博碩士論文知識加值系統。 <https://hdl.handle.net/11296/462243>
8. 渡部有隆, & Mirenkov, N. (2022). 會動的演算法：61個演算法動畫+全圖解逐步拆解，人工智慧、資料分析必備。旗標出版社。
9. 中央氣象署氣象資料開放平台。 <https://opendata.cwb.gov.tw/>
10. Flutter 官方網站範例。 <https://flutter.dev.org/tw/>
11. Steam教育學習網。Python Numpy 教學 - Array Create。 <https://steam.education/python-numpy>