



PEDESTRIAN OMAMORI

GET HOME SAFELY

113年1-5月道路交通事故統計

事故總件數

160,249

死亡人數(30日內)

1,193

受傷人數

214,124

主題統計

高齡者	死亡	478人
	受傷	30,360人
18-24歲 機車騎士	死亡	85人
	受傷	40,196人

縣市每十萬死傷人數

臺中市	1,364
臺南市	1,364
屏東縣	1,205

1位交通事故死亡的社會成本約 **1,600** 萬元

1位交通事故受傷民眾的社會成本約 **118** 萬元

2022年的交通事故社會總成本是 **6,380** 億

佔實質 GDP 的 **2.94%**

問題來了

我們可以做什麼降低行人交通事故發生的風險

- 一般民眾如何注意危險路段
- 如何特別關注老年人的安全需求



111年1-12月死亡 3,085人(每日 8.6人)，機車騎士死亡 1,954人，高齡者死亡 1,286人

年度 111年 ▼ 縣市 臺北市 ▼ 鄉鎮市區 全部 ▼ 計算方式 CBI ▼ 確定 操作說明



行人路徑危險熱點提醒

資料處理

整理行人交通
事故熱點數據



熱點地圖

使用地理資訊系統
呈現熱點地圖



路徑規劃

使用 Google Maps API
進行路徑規劃



家

將匹配完的地圖
和圖檔找個家



通知功能

使用LINE Bot
與使用者的互動流程





資料處理

搜尋

歷史交通事故資料：

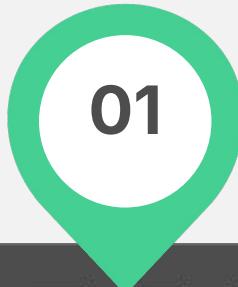
<https://data.gov.tw/dataset/12197>

<https://data.gov.twx/dataset/161199>

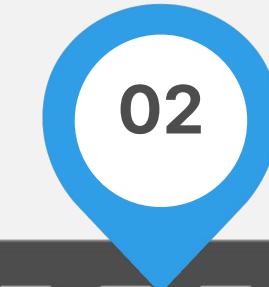
<https://data.gov.tw/dataset/167905>

肇事熱點地圖

:<https://roadsafety.tw/AccLocCbi>



檢視



統整篩選

抓行人取經緯



設 CBI 值

表 4.1-1 道路交通事故調查報告表 (一)

警察局名稱				總編號	轄區分局名稱	處理編號	道路交通事故調查報告表(一)			交通事故類別 (請打勾)	A1	A2	A3	
					分局									
①發生時間		②發生地點	(1)街道 地址	區(市)	里	路	段	巷	弄	號前	公尺處	③死傷人數		
□□年□□月□□日				鎮(鄉)	(村)	鄰	(街)	口	(東)	(南)	(西)	(北)	側	(附近)
□□時□□分星期□		(2)縣 (市)	(3)平交道名稱	路線	公里	公尺處	向	車道			24小時內	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
											2:30日內	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
④天候		⑤光線		⑥道路類別 (第1當事人)	⑦速限 (第1當事者)	⑧道路型態				⑨事故位置				
1 暴雨	1 日間自然光線	1 國道	<input type="checkbox"/>	(一)平交道	(三)單路部分	12 坡路		(一)交岔路口	08 慢車道	16 直線匝道				
2 強風	2 晨或暮光	2 省道	<input type="checkbox"/>	01 有遮斷器	06 隧道	13 巷弄		01 交岔路口內	09 一般車道	17 環道匝道				
3 風沙		3 縣道	<input type="checkbox"/>	02 無遮斷器	07 地下道	14 直路		02 交岔口附近	(未劃分快慢車道)	(四)其他				
4 霧或煙	3 夜間(或隧道、地下道、涵洞)有照明	4 鄉道	<input type="checkbox"/>	(二)交岔路	08 橋樑	15 其他		03 機車待轉區	10 公車專用道	18 行人穿越道				
5 雪		5 市區道路	<input type="checkbox"/>	03 三岔路	09 涵洞	(四)圓環廣場		04 機車停等區	11 機車專用道	19 穿越道附近				
6 雨	4 夜間(或隧道、地下道、涵洞)無照明	6 村里道路	<input type="checkbox"/>	04 四岔路	10 高架道路	16 圓環		(二)路段	12 機車優先道	20 人行道				
7 陰		7 專用道路	<input type="checkbox"/>	05 多岔路	11 弯曲路及附近	17 廣場		05 交通島	13 路肩、路緣	21 收費站附近				
8 晴		8 其他	<input type="checkbox"/>					(含楷化線)	06 迴轉道	22 其他				
									14 加速車道					
									15 減速車道					
⑩路面狀況				⑪道路障礙				⑫號誌				⑬車道劃分設施-分向設施		
(1)路面鋪裝		(2)路面狀態		(3)路面缺陷		(1)障礙物	(2)視距	(1)號誌種類	(2)號誌動作					
1 柏油	1 冰雪	1 路面鬆軟	1 道路工事(程)中	(一)不良		1 行車管制號誌		1 正常		(一)中央分向島	(三)單向禁止超車線			
2 水泥	2 油滑	2 突出(高低)不平	2 有堆積物	1 舊道		2 行車管制號誌		2 不正常		01 寬式	06 附標記			
3 碎石	3 泥濘	3 有坑洞	3 路上有停車	2 坡道		(附設行人專用號誌)		3 無動作		(50 公分以上)	07 無標記			
4 其他鋪裝	4 濕潤	4 無缺陷	4 其他障礙物	3 建築物		3 閃光號誌		4 無號誌		02 窄式附柵欄	(四)行車分向線			
5 無鋪裝	5 乾燥		5 無障礙物	4 樹木、農作物		4 無號誌				03 窄式無柵欄	08 附標記			
				5 路上停放車輛					04 附標記	09 無標記				
				6 其他					05 無標記	(五)無				
				(二)良好						10 無分向設施				
				7 良好										
⑭車道劃分設施-分道設施								⑮事故類型及型態						
(1)快車道或一般車道間		(2)快慢車道間		(3)路面邊線		(一)人與汽(機)車	(二)車與車	(三)汽(機)車本身		(四)平交道事故				
1 禁止變換車道線(附標記)		1 寬式快慢車道分隔島 (50 公分以上)		1. 有	01 對向通行中	10 對撞	18 路上翻車、摔倒	26 撞路樹、電桿	30 衝過(或撞壞)遮斷器					
2 禁止變換車道線(無標記)		2 窄式快慢車道分隔島(附柵欄)		2. 無	02 同向通行中	11 對向擦撞	19 衝出路外	27 撞動物	31 正越過平交道中					
3 車道線(附標記)		3 窄式快慢車道分隔島(無柵欄)			03 穿越道路中	12 同向擦撞	20 撞護欄(橋)	28 撞工程施工	32 暫停位置不當					
4 車道線(無標記)		4 快慢車道分隔線			04 在路上嬉戲	13 追撞	21 撞號誌、標誌桿	29 其他	33 在平交道內無法行動					
5 未繪設車道線		5 未繪設快慢車道分隔線			05 在路上作業中	14 倒車撞	22 撞收費亭	34 其他						
					06 街邊路中	15 路口交岔撞	23 撞交通島							
					07 從停車後(或中)穿出	16 側撞	24 撞非固定設施							
					08 佇立路邊(外)	17 其他	25 撞橋樑、建築物							

表 4.1-2 道路交通事故調查報告表 (二)

警察局名稱			總編號	轄區分局名稱	處理編號	道路交通事故調查報告表(二)-□									
				分局											
⑯當事者姓名		⑰屬(性)別	⑱身分證字號			⑲出生年月日	⑳住址					㉑電話	備註		
第幾當事者	1男														
	2女														
	3無或物(動物、堆置物)														
	4肇事逃逸尚未查獲														
㉒受傷程度	㉓主要傷處		㉔保護裝備		㉕行動電話		㉖當事者區分(類別)					㉗車輛牌照號碼			
1. 24 小時內死亡 2. 受傷 3. 未受傷 4. 不明 5. 2-30 日內死亡	01 頭部	07 手(腕)部	1 戴安全帽或繫安全帶(使用幼童安全椅) 2 未戴安全帽或未繫安全帶(未使用幼童安全椅) 3 不明 4 其他(行人、慢車駕駛人、汽車後座乘客)	1 未使用 2 使用手持 3 使用免持 4 不明 5 非汽(機)車駕駛人	(一)大客車 A01 公營公車 (四)半聯結車 B11 營業用 A03 公營客運 A04 民營客運 A05 遊覽車 (二)大貨車 A06 自用大客車 (六)小客車 A11 營業用 A12 自用 (三)全聯結車	A21 營業用	(七)小客車 A22 自用 (合客・貨兩用)	(九)軍車 D01 大客車 D02 軍用車	(十一)機車 F01 機車 F02 電動輔助自行車	G05 火車					
	02 頭部	08 臀(腰)部													
	03 胸部	09 多數傷													
	04 腹部	10 無													
	05 腰部	11 不明													
	06 背背部														
㉘車輛用途	㉙當事者行動狀態			㉚駕駛資格情形			㉛駕駛執照種類			㉜飲酒情形					
1 砂石車 2 幼童專用車 3 校車 4 施工特製車 5 教練車 6 裝載危險物品車 7 其他 8 非駕駛人及乘客	(一)車的狀態		08 向右變換車道	(二)人的狀態	01 起步	09 向前直行中	16 步行	1 有適當之駕照 2 無照(未達考照年齡) 3 無照(已達考照年齡) 4 超級駕駛 5 駕照被吊扣 6 駕照被吊銷 7 不明 8 非汽(機)車駕駛人	(一)駕照 01 聯結車 03 大貨車 05 聯結車 07 大貨車 09 駕照被吊扣 11 大型重型 13 非汽(機)車駕駛人	(四)軍用範例 02 大客車 04 小型車 06 大客車 08 小型車 10 駕照被吊銷 12 無駕駛執照 14 舊型	(九)軍車 12 大客車 14 小型車 15 國際(外國)駕照 16 其他駕照(證) 17 學習駕駛證 18 無駕駛執照 19 不明 20 非汽(機)車駕駛人	(十一)機車 F01 機車 F02 電動輔助自行車 F03 電動自行車 F04 人力車 F05 驢力車 F06 其他機車 F07 其他 F08 其他 F09 其他 F10 其他 F11 其他 F12 其他 F13 其他 F14 其他 F15 其他 F16 其他 F17 其他 F18 其他 F19 其他 F20 其他 F21 其他 F22 其他 F23 其他 F24 其他 F25 其他 F26 其他 F27 其他 F28 其他 F29 其他 F30 其他 F31 其他 F32 其他 F33 其他 F34 其他 F35 其他 F36 其他 F37 其他 F38 其他 F39 其他 F40 其他 F41 其他 F42 其他 F43 其他 F44 其他 F45 其他 F46 其他 F47 其他 F48 其他 F49 其他 F50 其他 F51 其他 F52 其他 F53 其他 F54 其他 F55 其他 F56 其他 F57 其他 F58 其他 F59 其他 F60 其他 F61 其他 F62 其他 F63 其他 F64 其他 F65 其他 F66 其他 F67 其他 F68 其他 F69 其他 F70 其他 F71 其他 F72 其他 F73 其他 F74 其他 F75 其他 F76 其他 F77 其他 F78 其他 F79 其他 F80 其他 F81 其他 F82 其他 F83 其他 F84 其他 F85 其他 F86 其他 F87 其他 F88 其他 F89 其他 F90 其他 F91 其他 F92 其他 F93 其他 F94 其他 F95 其他 F96 其他 F97 其他 F98 其他 F99 其他 F100 其他 F101 其他 F102 其他 F103 其他 F104 其他 F105 其他 F106 其他 F107 其他 F108 其他 F109 其他 F110 其他 F111 其他 F112 其他 F113 其他 F114 其他 F115 其他 F116 其他 F117 其他 F118 其他 F119 其他 F120 其他 F121 其他 F122 其他 F123 其他 F124 其他 F125 其他 F126 其他 F127 其他 F128 其他 F129 其他 F130 其他 F131 其他 F132 其他 F133 其他 F134 其他 F135 其他 F136 其他 F137 其他 F138 其他 F139 其他 F140 其他 F141 其他 F142 其他 F143 其他 F144 其他 F145 其他 F146 其他 F147 其他 F148 其他 F149 其他 F150 其他 F151 其他 F152 其他 F153 其他 F154 其他 F155 其他 F156 其他 F157 其他 F158 其他 F159 其他 F160 其他 F161 其他 F162 其他 F163 其他 F164 其他 F165 其他 F166 其他 F167 其他 F168 其他 F169 其他 F170 其他 F171 其他 F172 其他 F173 其他 F174 其他 F175 其他 F176 其他 F177 其他 F178 其他 F179 其他 F180 其他 F181 其他 F182 其他 F183 其他 F184 其他 F185 其他 F186 其他 F187 其他 F188 其他 F189 其他 F190 其他 F191 其他 F192 其他 F193 其他 F194 其他 F195 其他 F196 其他 F197 其他 F198 其他 F199 其他 F200 其他 F201 其他 F202 其他 F203 其他 F204 其他 F205 其他 F206 其他 F207 其他 F208 其他 F209 其他 F210 其他 F211 其他 F212 其他 F213 其他 F214 其他 F215 其他 F216 其他 F217 其他 F218 其他 F219 其他 F220 其他 F221 其他 F222 其他 F223 其他 F224 其他 F225 其他 F226 其他 F227 其他 F228 其他 F229 其他 F230 其他 F231 其他 F232 其他 F233 其他 F234 其他 F235 其他 F236 其他 F237 其他 F238 其他 F239 其他 F240 其他 F241 其他 F242 其他 F243 其他 F244 其他 F245 其他 F246 其他 F247 其他 F248 其他 F249 其他 F250 其他 F251 其他 F252 其他 F253 其他 F254 其他 F255 其他 F256 其他 F257 其他 F258 其他 F259 其他 F260 其他 F261 其他 F262 其他 F263 其他 F264 其他 F265 其他 F266 其他 F267 其他 F268 其他 F269 其他 F270 其他 F271 其他 F272 其他 F273 其他 F274 其他 F275 其他 F276 其他 F277 其他 F278 其他 F279 其他 F280 其他 F281 其他 F282 其他 F283 其 F284 其 F285 其 F286 其 F287 其 F288 其 F289 其 F290 其 F291 其 F292 其 F293 其 F294 其 F295 其 F296 其 F297 其 F298 其 F299 其 F300 其 F301 其 F302 其 F303 其 F304 其 F305 其 F306 其 F307 其 F308 其 F309 其 F310 其 F311 其 F312 其 F313 其 F314 其 F315 其 F316 其 F317 其 F318 其 F319 其 F320 其 F321 其 F322 其 F323 其 F324 其 F325 其 F326 其 F327 其 F328 其 F329 其 F330 其 F331 其 F332 其 F333 其 F334 其 F335 其 F336 其 F337 其 F338 其 F339 其 F340 其 F341 其 F342 其 F343 其 F344 其 F345 其 F346 其 F347 其 F348 其 F349 其 F350 其 F351 其 F352 其 F353 其 F354 其 F355 其 F356 其 F357 其 F358 其 F359 其 F360 其 F361 其 F362 其 F363 其 F364 其 F365 其 F366 其 F367 其 F368 其 F369 其 F370 其 F371 其 F372 其 F373 其 F374 其 F375 其 F376 其 F377 其 F378 其 F379 其 F380 其 F381 其 F382 其 F383 其 F384 其 F385 其 F386 其 F387 其 F388 其 F389 其 F390 其 F391 其 F392 其 F393 其 F394 其 F395 其 F396 其 F397 其 F398 其 F399 其 F400 其 F401 其 F402 其 F403 其 F404 其 F405 其 F406 其 F407 其 F408 其 F409 其 F410 其 F411 其 F412 其 F413 其 F414 其 F415 其 F416 其 F417 其 F418 其 F419 其 F420 其 F421 其 F422 其 F423 其 F424 其 F425 其 F426 其 F427 其 F428 其 F429 其 F430 其 F431 其 F432 其 F433 其 F434 其 F435 其 F436 其 F437 其 F438 其 F439 其 F440 其 F441 其 F442 其 F443 其 F444 其 F445 其 F446 其 F447 其 F448 其 F449 其 F450 其 F451 其 F452 其 F453 其 F454 其 F455 其 F456 其 F457 其 F458 其 F459 其 F460 其 F461 其 F462 其 F463 其 F464 其 F465 其 F466 其 F467 其 F468 其 F469 其 F470 其 F471 其 F472 其 F473 其 F474 其 F475 其 F476 其 F477 其 F478 其 F479 其 F480 其 F481 其 F482 其 F483 其 F484 其 F485 其 F486 其 F487 其 F488 其 F489 其 F490 其 F491 其 F492 其 F493 其 F494 其 F495 其 F496 其 F497 其 F498 其 F499 其 F500 其 F501 其 F502 其 F503 其 F504 其 F505 其 F506 其 F507 其 F508 其 F509 其 F510 其 F511 其 F512 其 F513 其 F514 其 F515 其 F516 其 F517 其 F518 其 F519 其 F520 其 F521 其 F522 其 F523 其 F524 其 F525 其 F526 其 F527 其 F528 其 F529 其 F530 其 F531 其 F532 其 F533 其 F534 其 F535 其 F536 其 F537 其 F538 其 F539 其 F540 其 F541 其 F542 其 F543 其 F544 其 F545 其 F546 其 F547 其 F548 其 F549 其 F550 其 F551 其 F552 其 F553 其 F554 其 F555 其 F556 其 F557 其 F558 其 F559 其 F560 其 F561 其 F562 其 F563 其 F564 其 F565 其 F566 其 F567 其 F568 其 F569 其 F570 其 F571 其 F572 其 F573 其 F574 其 F575 其 F576 其 F577 其 F578 其 F579 其 F580 其 F581 其 F582 其 F583 其 F584 其 F585 其 F586 其 F587 其 F588 其 F589 其 F590 其 F591 其 F592 其 F593 其 F594 其 F595 其 F596 其 F597 其 F598 其 F599 其 F600 其 F601 其 F602 其 F603 其 F604 其 F605 其 F606 其 F607 其 F608 其 F609 其 F610 其 F611 其 F612 其 F613 其 F614 其 F615 其 F616 其 F617 其 F618 其 F619 其 F620 其 F621 其 F622 其 F623 其 F624 其 F625 其 F626 其 F627 其 F628 其 F629 其 F630 其 F631 其 F632 其 F633 其 F634 其 F635 其 F636 其 F637 其 F638 其 F639 其 F640 其 F641 其 F642 其 F643 其 F644 其 F645 其 F646 其 F647 其 F648 其 F649 其 F650 其 F651 其 F652 其 F653 其 F654 其 F655 其 F656 其 F657 其 F658 其 F659 其 F660 其 F661 其 F662 其 F663 其 F664 其 F665 其 F666 其 F667 其 F668 其 F669 其 F670 其 F671 其 F672 其 F673 其 F674 其 F6			

編號	經度	緯度	事件類別	CBI值
108_A1_1	121.547207	24.896801	A1	3
108_A1_2	121.712519	25.097169	A1	3
108_A1_3	120.290971	23.034979	A1	3
108_A1_4	121.704939	25.141623	A1	3
108_A1_5	120.928382	24.683937	A1	3
108_A1_6	121.252581	24.939473	A1	3
108_A1_7	120.807731	24.571506	A1	3
108_A1_8	121.484635	25.076559	A1	3
108_A1_9	121.079359	24.724161	A1	3
108_A1_10	120.333588	22.605671	A1	3
108_A1_11	121.413873	24.955711	A1	3
108_A1_12	120.647394	24.43418	A1	3
108_A1_13	120.774332	23.941834	A1	3
108_A1_14	121.510881	25.080026	A1	3
108_A1_15	121.42639	24.990655	A1	3
108_A1_16	120.67529	24.09779	A1	3
108_A1_17	121.281663	24.872455	A1	3
108_A1_18	120.668893	24.132052	A1	3

抓行人
取經緯
CBI 死亡3 受傷1
108-102年 85,372筆

行人路徑危險熱點提醒

資料處理

整理行人交通事故熱點數據



熱點地圖

使用地理資訊系統
呈現熱點地圖



路徑規劃

使用 Google Maps API
進行路徑規劃



家

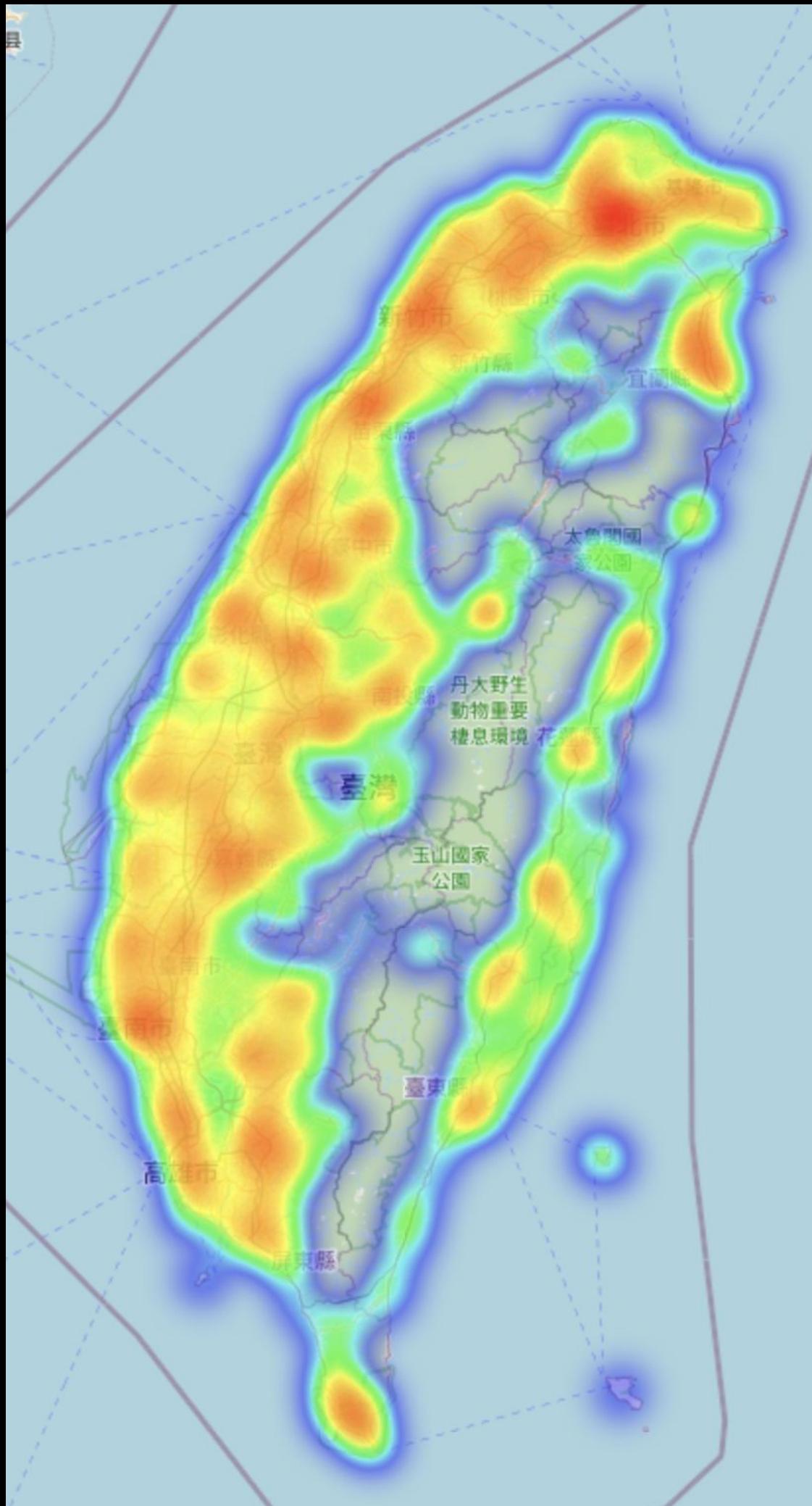
將匹配完的地圖
和圖檔找個家



通知功能

使用LINE Bot
與使用者的互動流程





```
#· 加載和清理數據  
檔案路徑·=·'./Users/chinlin/Desktop/test/0_ALL_108-112_CBI.csv'··#·您的CSV檔案路徑  
資料框·=·pd.read_csv(檔案路徑)  
清理後資料框·=·資料框.dropna(subset=['經度',·'緯度'])  
清理後資料框['CBI值']·=·清理後資料框['事件類別'].apply(lambda·x:·3·if·x==·'A1'·else·1)  
地理資料框·=·gpd.GeoDataFrame(清理後資料框,·geometry=gpd.points_from_xy(清理後資料框.經度,·清理後資料框.緯度))  
  
#· Haversine公式來計算兩個地理座標之間的距離  
def·haversine(緯度1,·經度1,·緯度2,·經度2):  
    ···地球半徑·=·6371··#·地球半徑 (公里)  
    ···緯度差·=·radians(緯度2··-·緯度1)  
    ···經度差·=·radians(經度2··-·經度1)  
    ···a·=·sin(緯度差·/·2)·**·2·+·cos(radians(緯度1))·*·cos(radians(緯度2))·*·sin(經度差·/·2)·**·2  
    ···c·=·2·*·atan2(sqrt(a),·sqrt(1··-·a))  
    ···return·地球半徑·*·c  
  
#· 生成熱力圖  
def·生成熱力圖(地理資料框):  
    ···#·確定地圖中心  
    ···地圖中心緯度·=·地理資料框['緯度'].mean()  
    ···地圖中心經度·=·地理資料框['經度'].mean()  
    ···地圖·=·folium.Map(location=[地圖中心緯度,·地圖中心經度],·zoom_start=12)  
  
    ···#·準備熱力圖數據  
    ···熱力圖數據·=·[[row['geometry'].y,·row['geometry'].x,·row['CBI值']]·for·index,·row·in·地理資料框.iterrows()]  
    ···  
    ···#·添加熱力圖  
    ···HeatMap(熱力圖數據,·radius=15).add_to(地圖)  
    ···  
    ···#·保存熱力圖到HTML檔案  
    ···地圖HTML檔案路徑·=·'./Users/chinlin/Desktop/test/交通事故熱力圖.html'  
    ···地圖.save(地圖HTML檔案路徑)  
    ···print(f"熱力圖已生成並保存到·{地圖HTML檔案路徑}")  
  
if·__name__·==·"__main__":  
    ···生成熱力圖(地理資料框)
```

行人路徑危險熱點提醒

資料處理

整理行人交通
事故熱點數據



熱點地圖

使用地理資訊系統
呈現熱點地圖



路徑規劃

使用 Google Maps API
進行路徑規劃



家

將匹配完的地圖
和圖檔找個家



通知功能

使用LINE Bot
與使用者的互動流程



路徑規 劃



修改路徑

解碼Google Maps的折線數據
將其一系列經緯度點進行修正

路徑與交通事故熱點匹配

與已知的交通事故熱點進行距離計算
篩選出距離路徑一定範圍內的事故點

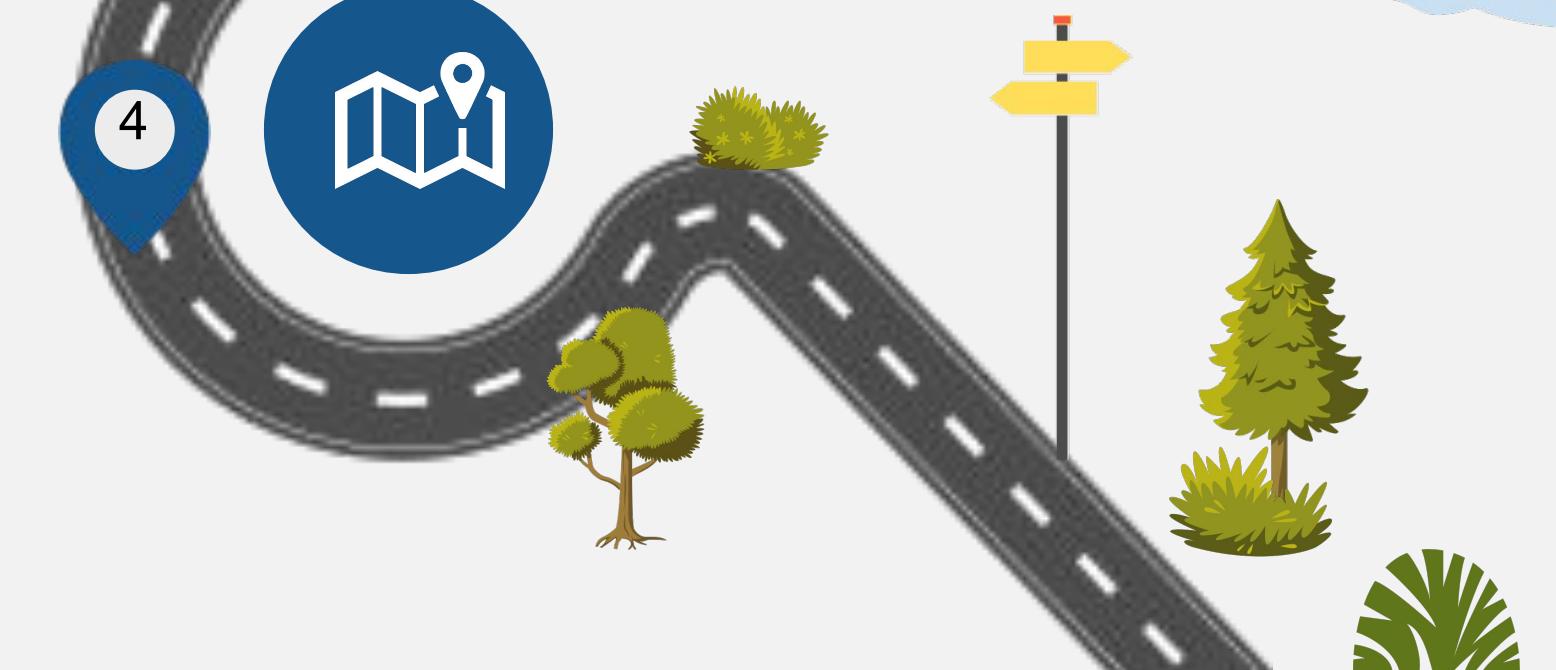
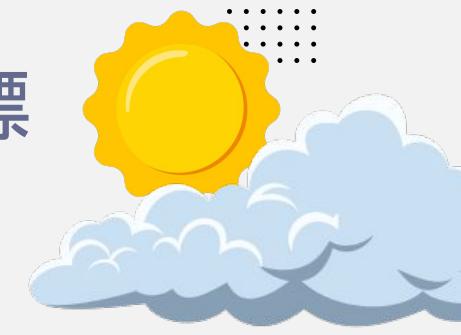


變成一條線

生成路徑後將這條路徑
轉換成一條直線

Google Maps API

我不知道他在幹嘛亂走亂標



```

import requests
from shapely.geometry import LineString
-----
# Google Maps API Key
api_key = 'YOUR_GOOGLE_MAPS_API_KEY'

def get_route(start, end):
    """
    使用Google Maps API生成步行路徑
    """
    base_url = "https://maps.googleapis.com/maps/api/directions/json"
    params = {
        "origin": f"{start[0]},{start[1]}",
        "destination": f"{end[0]},{end[1]}",
        "mode": "walking",
        "key": api_key
    }
    response = requests.get(base_url, params=params)
    data = response.json()

    if data['status'] == 'OK':
        polyline = data['routes'][0]['overview_polyline']['points']
        return decode_polyline(polyline)
    else:
        raise Exception(f"Error fetching route: {data['status']}")

```

```

def decode_polyline(polyline_str):
    """
    解碼Google Maps的折線數據，將其轉化為一系列經緯度點
    """
    index, lat, lng = 0, 0, 0
    coordinates = []

    while index < len(polyline_str):
        shift, result = 0, 0
        while True:
            byte = ord(polyline_str[index]) - 63
            index += 1
            result |= (byte & 0x1f) << shift
            shift += 5
            if not (byte >= 0x20):
                break
        dlat = (~result >> 1) if (result & 1) else (result >> 1)
        lat += dlat
        shift, result = 0, 0
        while True:
            byte = ord(polyline_str[index]) - 63
            index += 1
            result |= (byte & 0x1f) << shift
            shift += 5
            if not (byte >= 0x20):
                break
        dlng = (~result >> 1) if (result & 1) else (result >> 1)
        lng += dlng
        coordinates.append((lat / 1e5, lng / 1e5))
    return coordinates

```

```

from shapely.geometry import Point
import geopandas as gpd
from math import radians, sin, cos, sqrt, atan2

# 加載交通事故熱點數據
file_path = '/Users/chinlin/Desktop/test/0_ALL_108-112_CBI.csv'
df = gpd.read_file(file_path)
df = df.dropna(subset=['經度', '緯度'])
gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.經度, df.緯度))

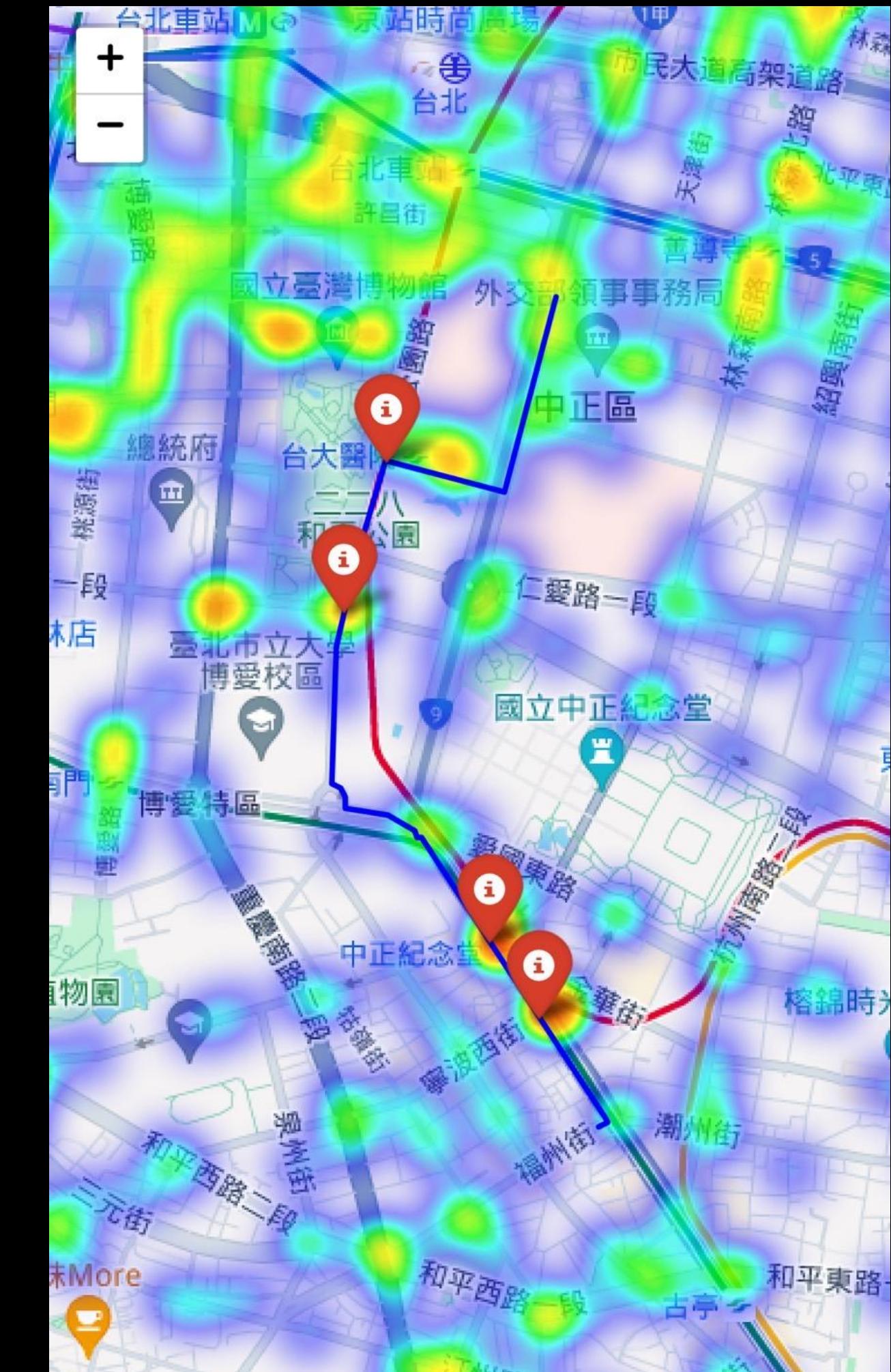
def haversine(lat1, lon1, lat2, lon2):
    """
    Haversine公式計算兩個地點之間的距離
    """
    R = 6371 # 地球半徑 (公里)
    dlat = radians(lat2 - lat1)
    dlon = radians(lon2 - lon1)
    a = sin(dlat/2)**2 + cos(radians(lat1)) * cos(radians(lat2)) * sin(dlon/2)**2
    c = 2 * atan2(sqrt(a), sqrt(1-a))
    return R * c

def match_route_with_hotspots(route_points):
    """
    將生成的路徑與事故熱點匹配
    """
    route_line = LineString(route_points)
    hotspots = []

    for idx, row in gdf.iterrows():
        distance = route_line.distance(row.geometry)
        if distance < 0.1: # 距離小於100米
            hotspots.append((row.geometry.y, row.geometry.x))

    return hotspots

```



行人路徑危險熱點提醒

資料處理

整理行人交通
事故熱點數據



熱點地圖

使用地理資訊系統
呈現熱點地圖



路徑規劃

使用 Google Maps API
進行路徑規劃



家

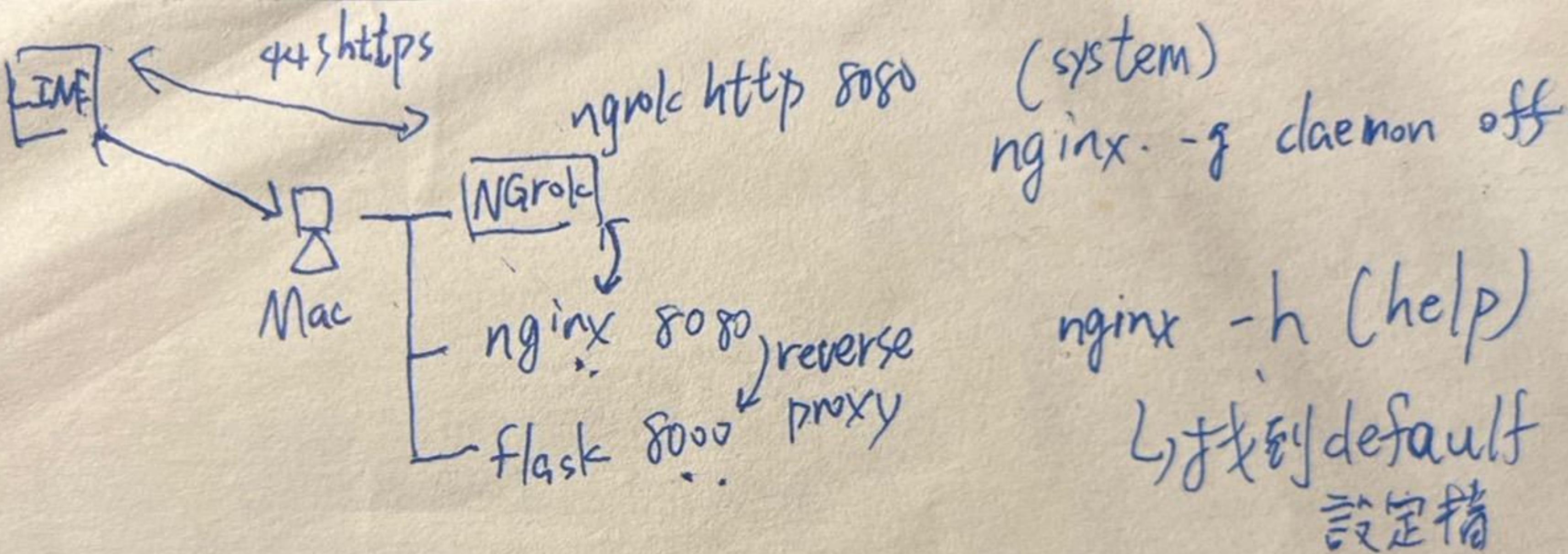
將匹配完的地圖
和圖檔找個家



通知功能

使用LINE Bot
與使用者的互動流程





`tail -f error.log`

(system)
`nginx -g daemon off`

`nginx -h (help)`

↳ 找到 default
 設定檔

`nginx.conf`

`nginx -t (設定檔測試)`

`nginx -s reload`

```
}
```

```
location /static/ {
    alias /Users/chinlin/Desktop/test/templates/static/; # 靜態文件的黃金路徑
}
```

```
location =/favicon.ico {
    access_log off;
    log_not_found off;
}
```

```
}
```

```
Last login: Mon Aug 12 14:35:24 on ttys013
chinlin@ChindeMacBook-Air ~ % nginx -t
```

```
nginx: the configuration file /opt/homebrew/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /opt/homebrew/etc/nginx/nginx.conf test is successful
chinlin@ChindeMacBook-Air ~ % nginx -s reload
chinlin@ChindeMacBook-Air ~ % █
```

行人路徑危險熱點提醒

資料處理

整理行人交通
事故熱點數據



熱點地圖

使用地理資訊系統
呈現熱點地圖



路徑規劃

使用 Google Maps API
進行路徑規劃



家

將匹配完的地圖
和圖檔找個家



通知功能

使用LINE Bot
與使用者的互動流程



通知功能



```

#.建立RichMenu
def create_rich_menu():
    # 調整RichMenu的大小，將高度設置為843
    rich_menu_to_create = RichMenu(
        size=RichMenuSize(width=2500, height=843),
        selected=False,
        name="RichMenu",
        chat_bar_text="確認路徑是否有交通事故熱點",
        areas=[
            RichMenuArea(
                bounds=RichMenuBounds(x=0, y=0, width=1250, height=843),
                action=MessageAction(label="開始", text="請傳送您的起點")
            ),
            RichMenuArea(
                bounds=RichMenuBounds(x=1250, y=0, width=1250, height=843),
                action=MessageAction(label="結束", text="請傳送您的終點")
            )
        ]
    )

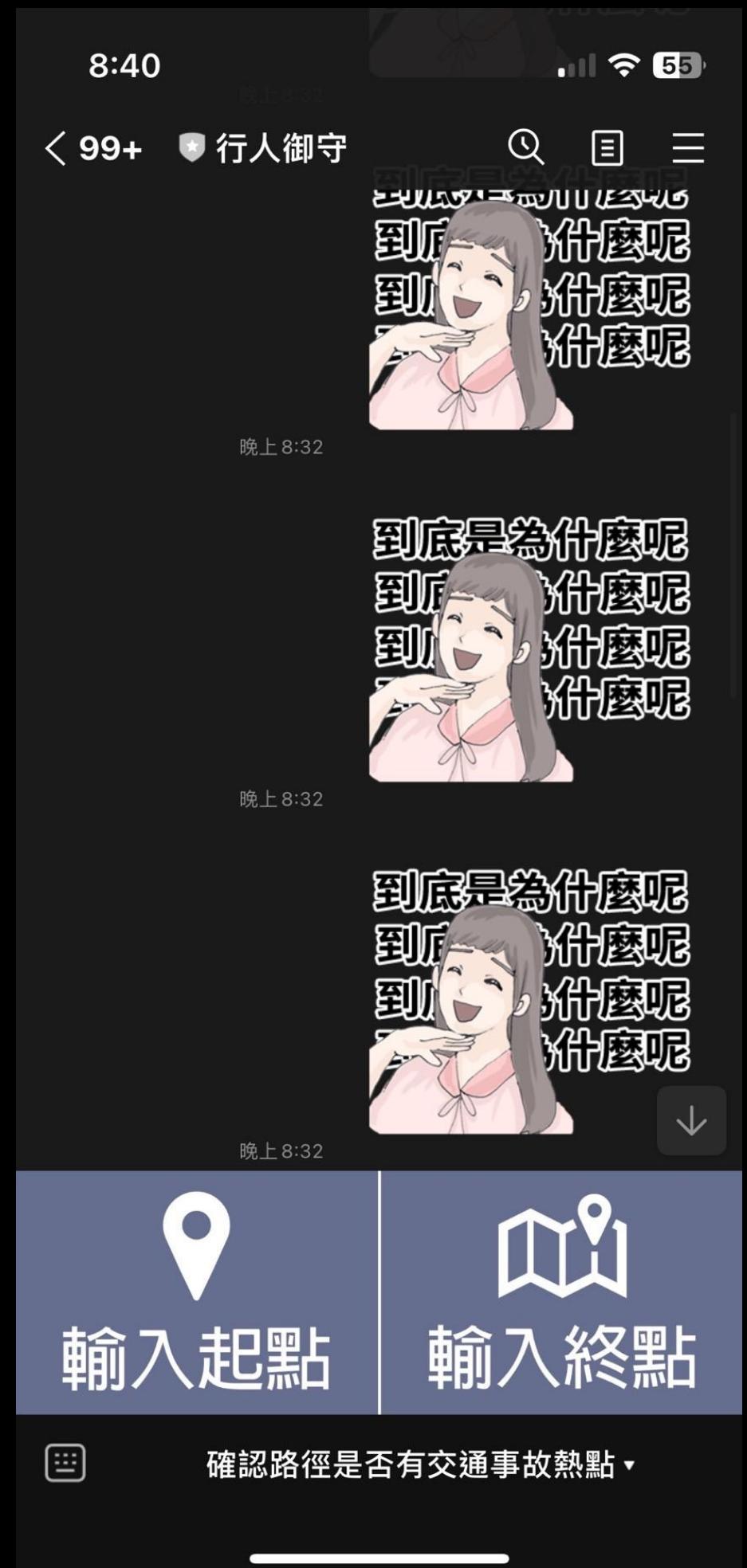
    # 創建並設置RichMenu
    rich_menu_id = line_bot_api.create_rich_menu(rich_menu=rich_menu_to_create)

    with open("/Users/chinlin/Desktop/test/20240812_6.jpg", "rb") as f:
        line_bot_api.set_rich_menu_image(rich_menu_id, "image/jpeg", f)

    line_bot_api.set_default_rich_menu(rich_menu_id)

create_rich_menu()

```



```

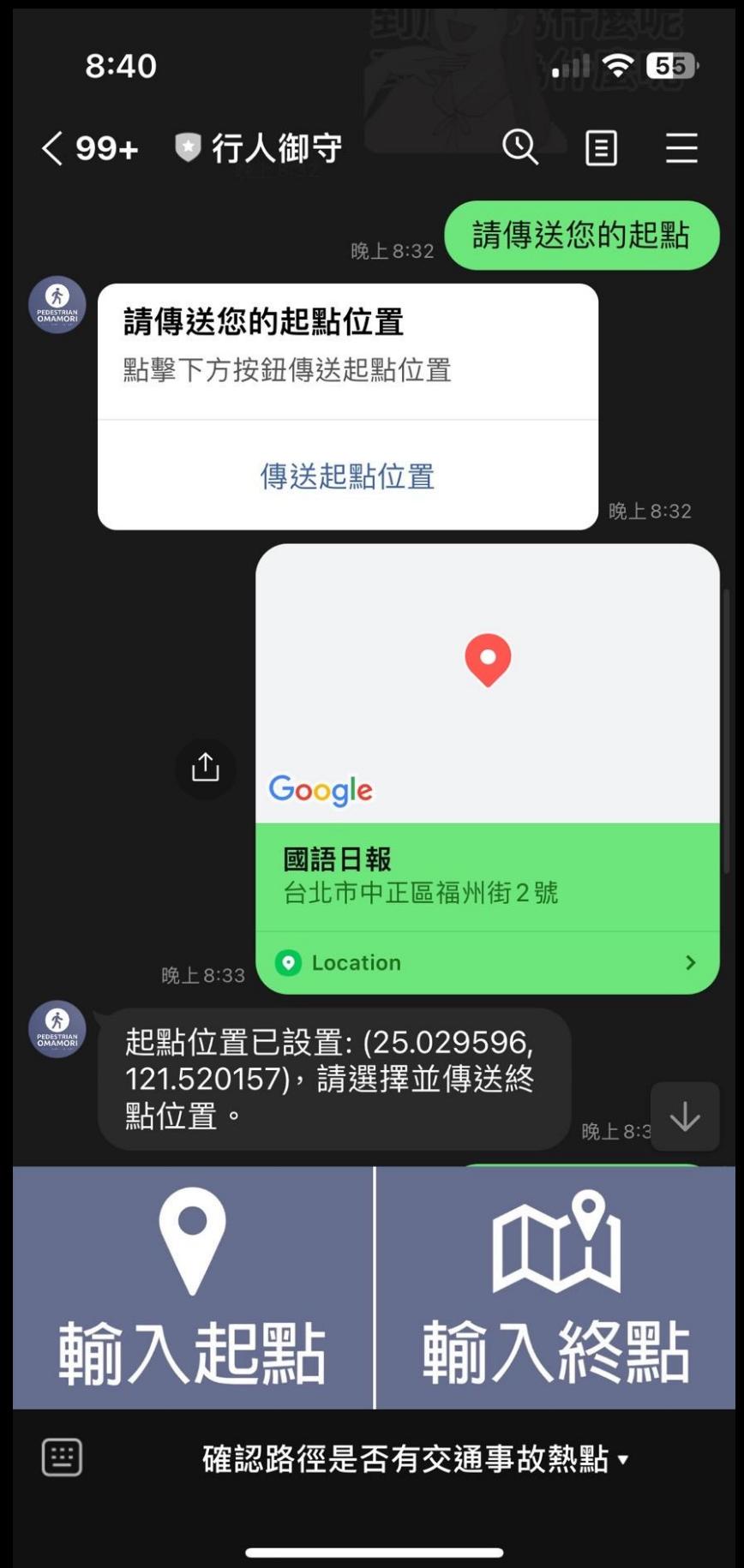
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    try:
        user_id = event.source.user_id
        app.logger.info(f"User ID: {user_id}") # 打印用戶ID

        if event.message.text == "請傳送您的起點":
            start_buttons_template = ButtonsTemplate(
                title='請傳送您的起點位置',
                text='點擊下方按鈕傳送起點位置',
                actions=[LocationAction(label='傳送起點位置')])
            template_message = TemplateSendMessage(alt_text='請傳送您的起點位置', template=start_buttons_template)
            line_bot_api.reply_message(event.reply_token, template_message)

        elif event.message.text == "請傳送您的終點":
            end_buttons_template = ButtonsTemplate(
                title='請傳送您的終點位置',
                text='點擊下方按鈕傳送終點位置',
                actions=[LocationAction(label='傳送終點位置')])
            template_message = TemplateSendMessage(alt_text='請傳送您的終點位置', template=end_buttons_template)
            line_bot_api.reply_message(event.reply_token, template_message)

    except Exception as e:
        app.logger.error(f"Error handling message: {e}")
        line_bot_api.reply_message(
            event.reply_token,
            TextMessage(text=f"Error: {e}"))

```



```

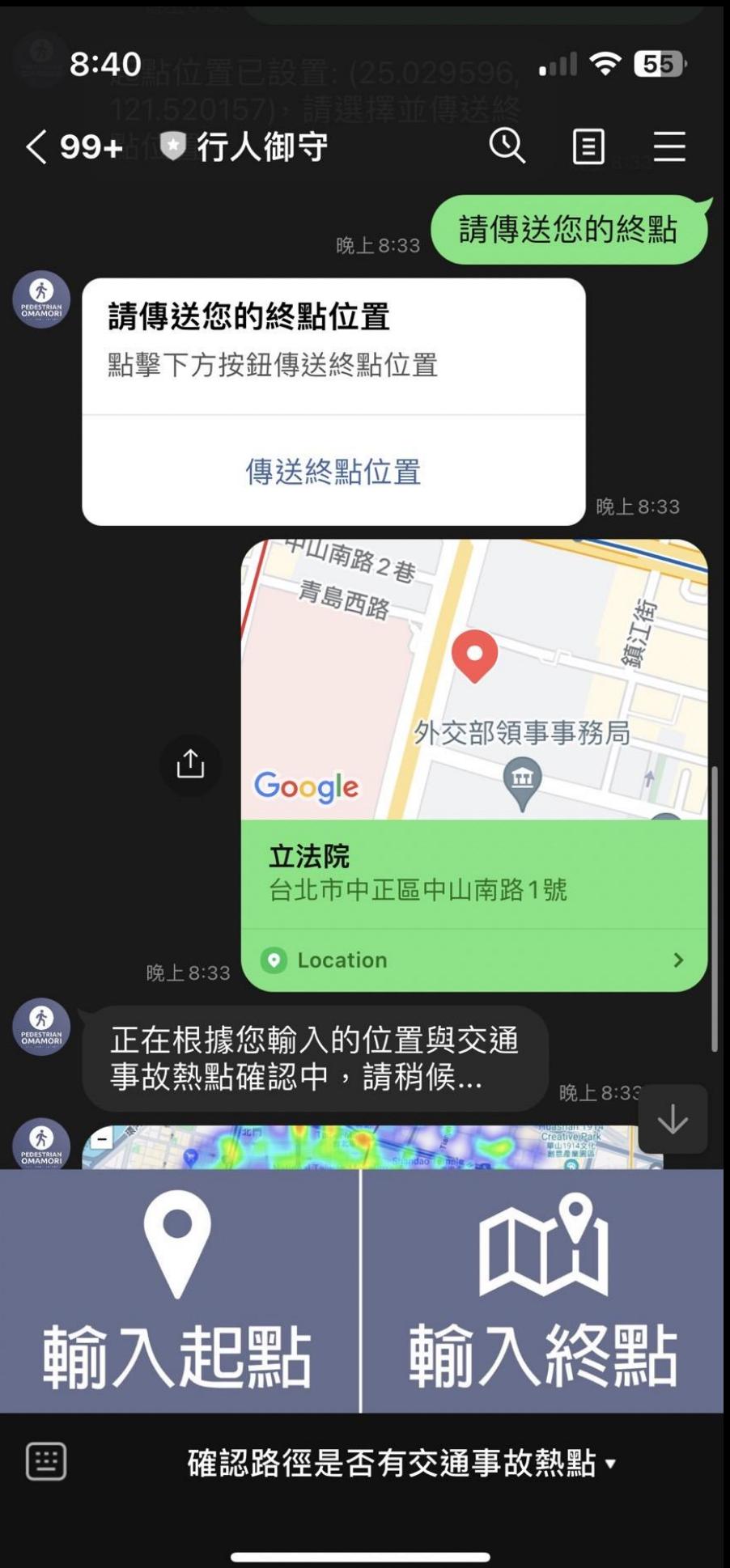
@handler.add(MessageEvent, message=LocationMessage)
def handle_location_message(event):
    user_id = event.source.user_id
    location = (event.message.latitude, event.message.longitude)

    with lock:
        if user_locations[user_id]['start'] is None:
            user_locations[user_id]['start'] = location
            app.logger.info(f"Set start location for user {user_id}: {location}")
            line_bot_api.reply_message(
                event.reply_token,
                TextSendMessage(text=f"起點位置已設置: {location}，請選擇並傳送終點位置。"))
        elif user_locations[user_id]['end'] is None:
            user_locations[user_id]['end'] = location
            start_location = user_locations[user_id]['start']
            end_location = user_locations[user_id]['end']

            line_bot_api.push_message(
                user_id,
                TextSendMessage(text="正在根據您輸入的位置與交通事故熱點確認中，請稍候..."))

        user_locations[user_id] = {'start': None, 'end': None}
        thread = threading.Thread(target=send_route_plan, args=(start_location, end_location, user_id))
        thread.start()

```



```

flex_message = {
    "type": "bubble",
    "hero": {
        "type": "image",
        "url": f'https://d669-123-51-152-222.ngrok-free.app/static/{os.path.basename(map_png_path)}',
        "size": "full",
        "aspectRatio": "20:13",
        "aspectMode": "cover",
        "action": {
            "type": "uri",
            "uri": f'https://d669-123-51-152-222.ngrok-free.app/static/{os.path.basename(map_html_path)}'
        }
    },
    "body": {
        "type": "box",
        "layout": "vertical",
        "contents": [
            {
                "type": "text",
                "text": "走路不要滑手機",
                "weight": "bold",
                "size": "xl"
            },
            {
                "type": "button",
                "style": "primary",
                "action": {
                    "type": "uri",
                    "label": "查看完整地圖",
                    "uri": f'https://d669-123-51-152-222.ngrok-free.app/static/{os.path.basename(map_html_path)}'
                }
            }
        ]
    }
}

line_bot_api.push_message(user_id, FlexSendMessage(alt_text="路徑地圖已生成", contents=flex_message))

```



```

def send_route_plan(start_location, end_location, user_id):
    try:
        timestamp = datetime.now().strftime("%Y%m%d%H%M%S")
        unique_filename = f"{user_id}_{timestamp}"

        route = get_route(start_location, end_location, api_key)

        route_points = [(lat, lng) for lat, lng in route]
        route_line = LineString(route_points)

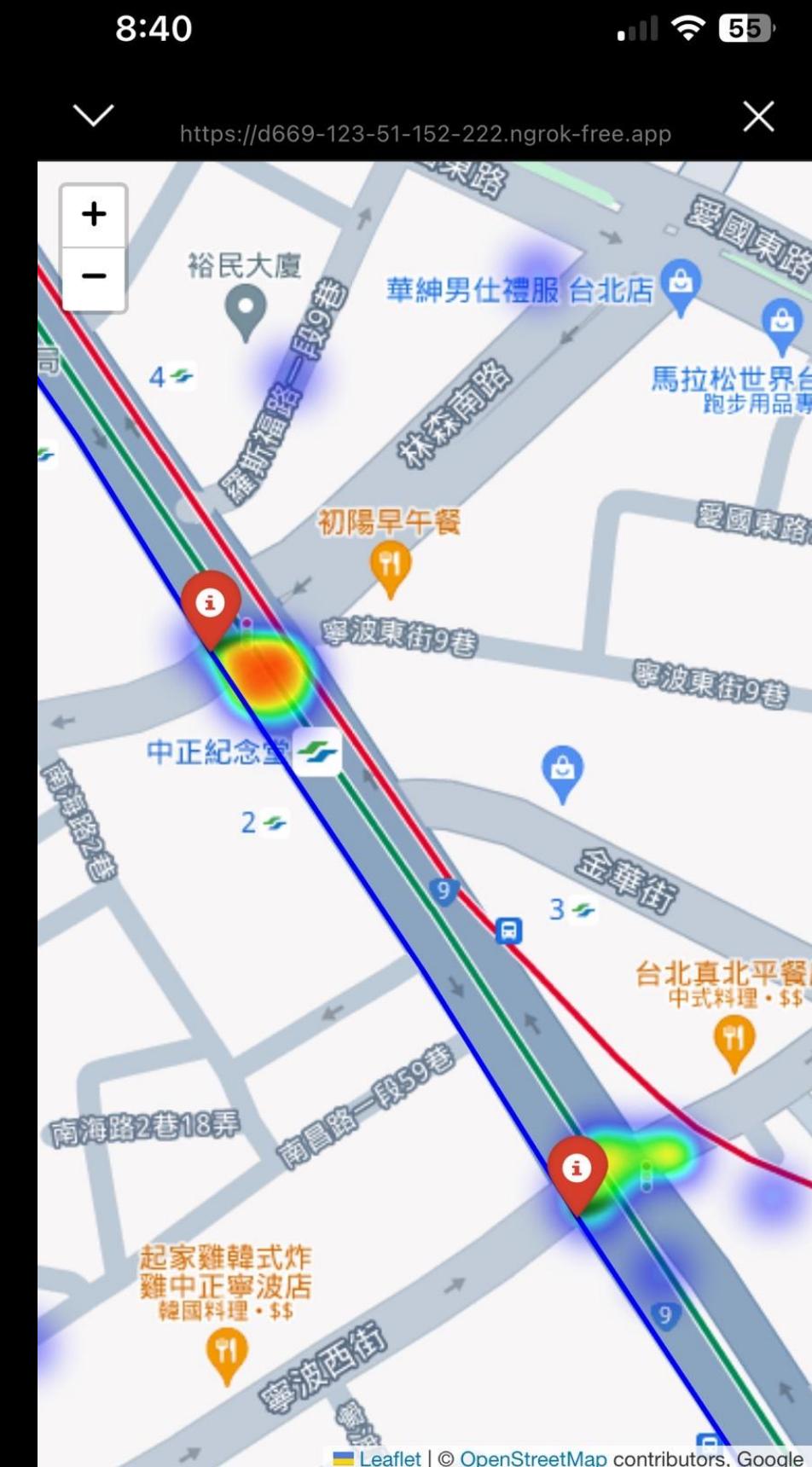
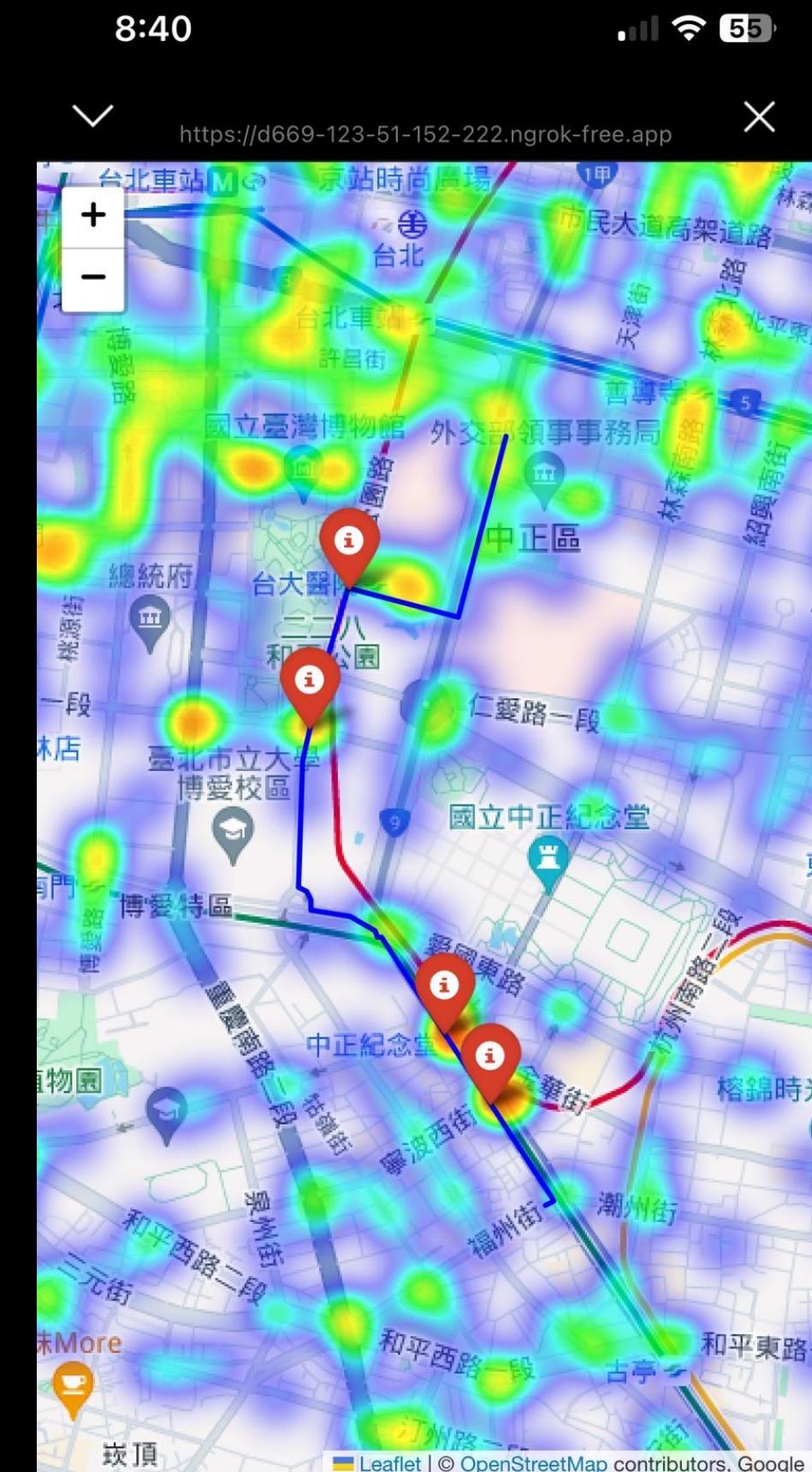
        latitudes = [point[0] for point in route_points]
        longitudes = [point[1] for point in route_points]
        route_center = (sum(latitudes) / len(latitudes), sum(longitudes) / len(longitudes))

        lat_span = max(latitudes) - min(latitudes)
        lon_span = max(longitudes) - min(longitudes)
        max_span = max(lat_span, lon_span)

        if max_span < 0.005:
            zoom = 18
        elif max_span < 0.01:
            zoom = 16
        elif max_span < 0.05:
            zoom = 14
        elif max_span < 0.1:
            zoom = 12
        elif max_span < 0.5:
            zoom = 10
        else:
            zoom = 8

        m = folium.Map(location=route_center, zoom_start=zoom)
        m.fit_bounds([[min(latitudes), min(longitudes)], [max(latitudes), max(longitudes)]])

```



```

# 接收來自Android應用的定位數據
@app.route('/', methods=['POST'])
def receive_location():
    data = request.json
    latitude = data.get('latitude')
    longitude = data.get('longitude')

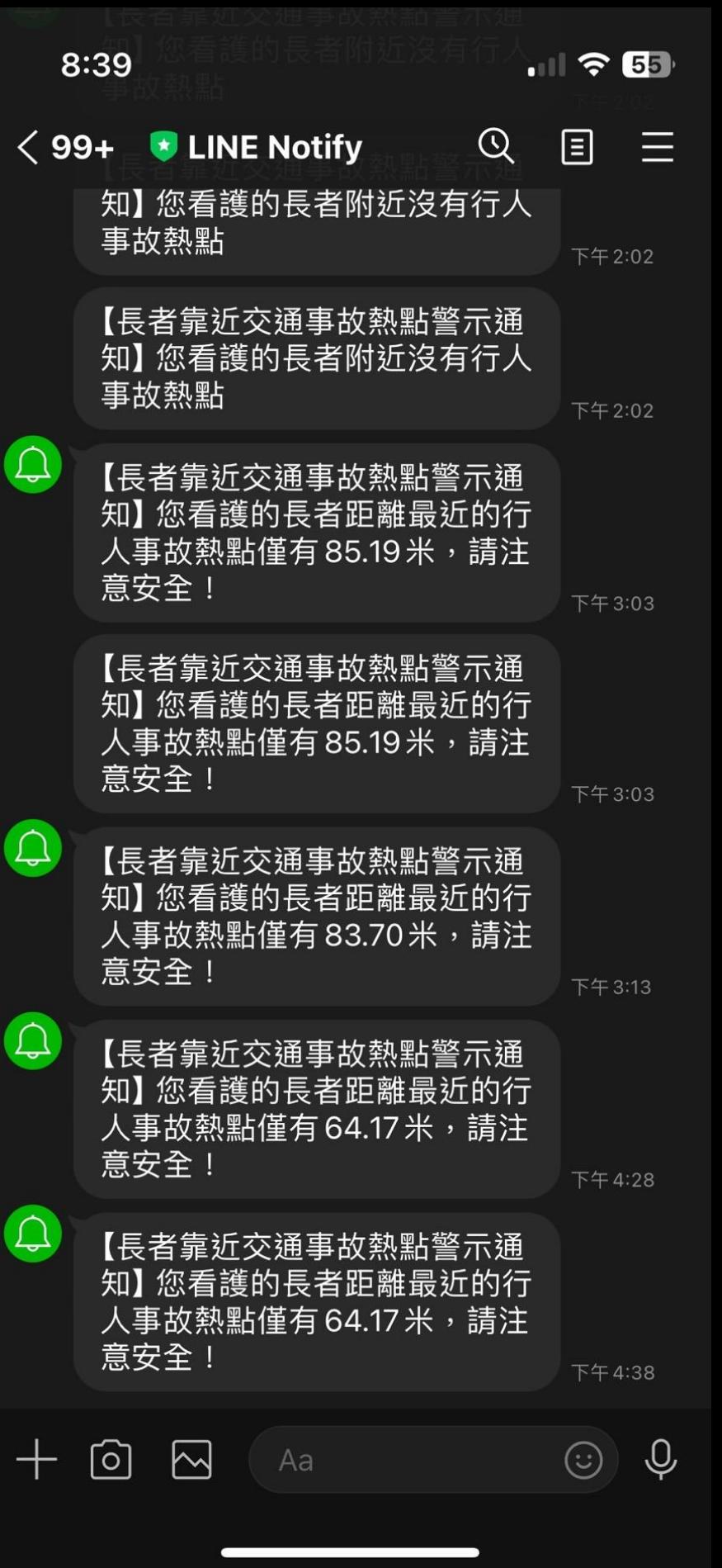
    if latitude is None or longitude is None:
        return jsonify({"status": "error", "message": "Missing latitude or longitude"}), 400

    try:
        latitude = float(latitude)
        longitude = float(longitude)
    except ValueError:
        return jsonify({"status": "error", "message": "Invalid latitude or longitude format"}), 400

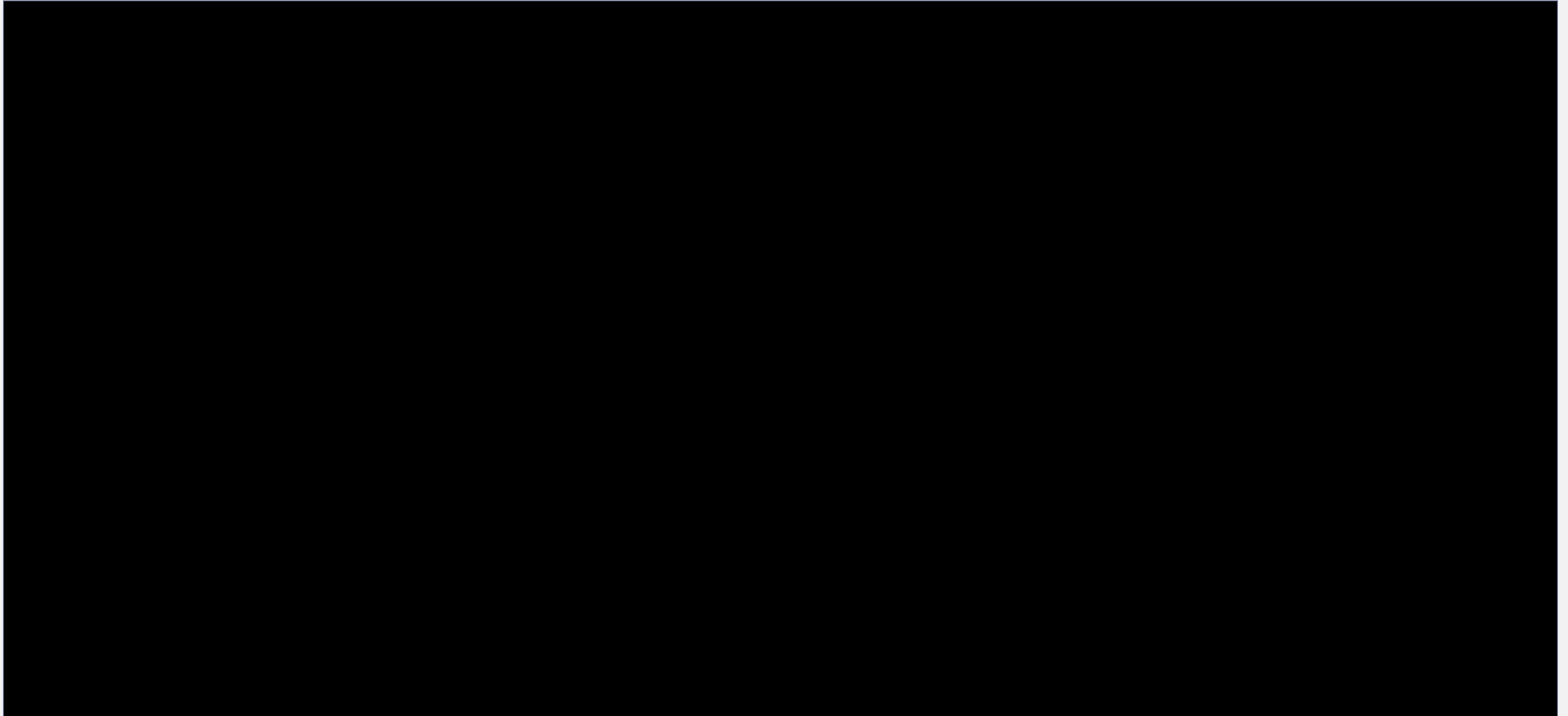
    # 檢查用戶位置與事故熱點的距離
    for idx, row in gdf.iterrows():
        distance_km = haversine(latitude, longitude, row.geometry.y, row.geometry.x)
        if distance_km <= 0.1: # 假設100米作為閾值
            message = f"您看護的長者距離最近的行人事故熱點僅有{distance_km * 1000:.2f}米，請注意安全！"
            send_line_notify(message)
            return jsonify({"status": "warning", "message": message}), 200

    message = f"您看護的長者附近沒有行人事故熱點"
    send_line_notify(message)
    return jsonify({"status": "safe", "message": "您看護的長者附近沒有行人事故熱點"}), 200

```



Demo

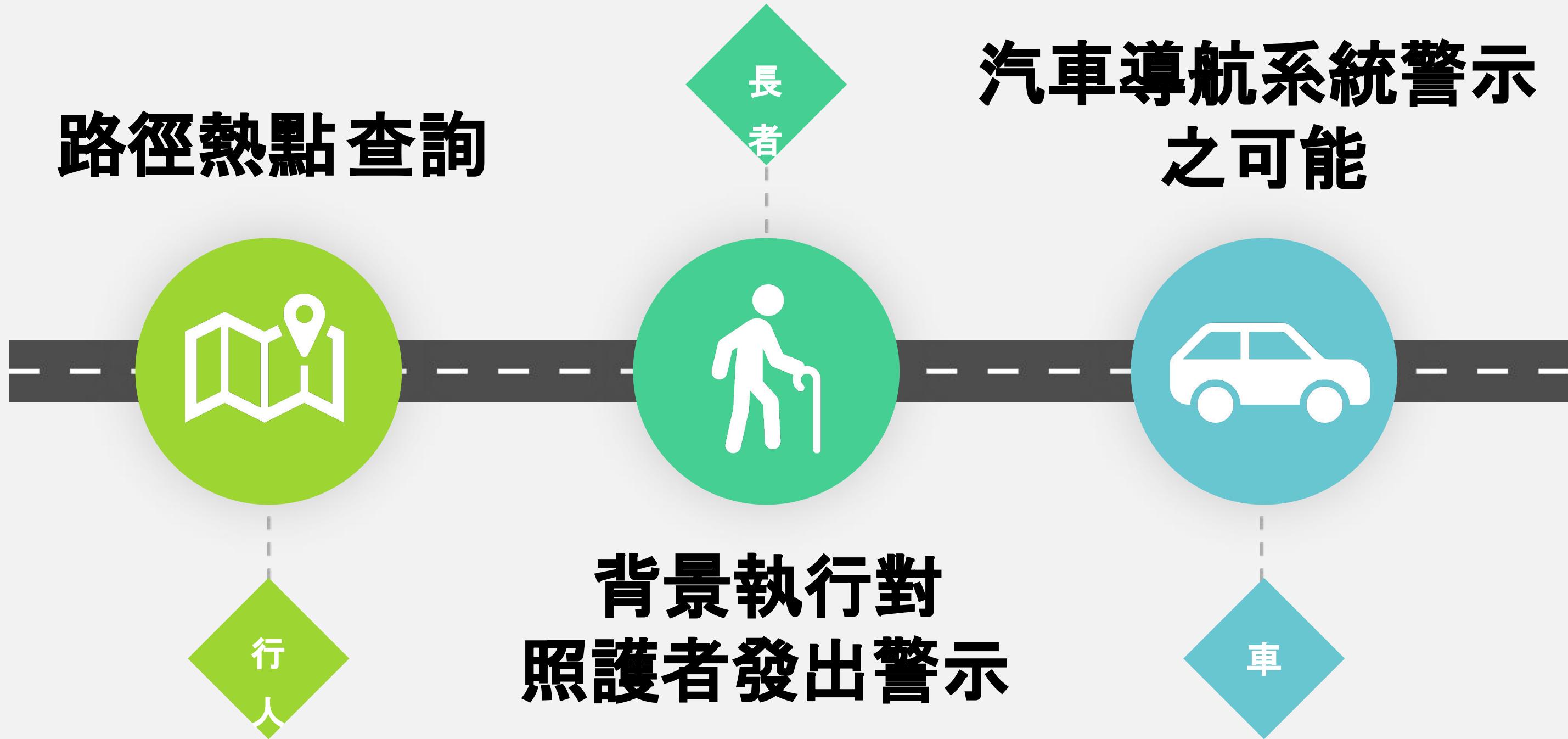


交通事件屬於高度隨機的稀有事件 專題的實益是？

路徑熱點查詢

汽車導航系統警示
之可能

背景執行對
照護者發出警示



is_hotspot

True 918

False 6

Name: count, dtype: int64

Optimization terminated successfully.

Current function value: 0.037281

Iterations 23

Logit Regression Results

Dep. Variable:	is_hotspot	No. Observations:	924
Model:	Logit	Df Residuals:	921
Method:	MLE	Df Model:	2
Date:	Tue, 09 Jul 2024	Pseudo R-squ.:	0.04846
Time:	11:08:38	Log-Likelihood:	-34.448
converged:	True	LL-Null:	-36.202
Covariance Type:	nonrobust	LLR p-value:	0.1730

	coef	std err	z	P> z	[0.025	0.975]
const	2833.4185	1494.388	1.896	0.058	-95.527	5762.364
經度	-22.0615	11.648	-1.894	0.058	-44.890	0.767
緯度	-5.8563	13.475	-0.435	0.664	-32.266	20.553

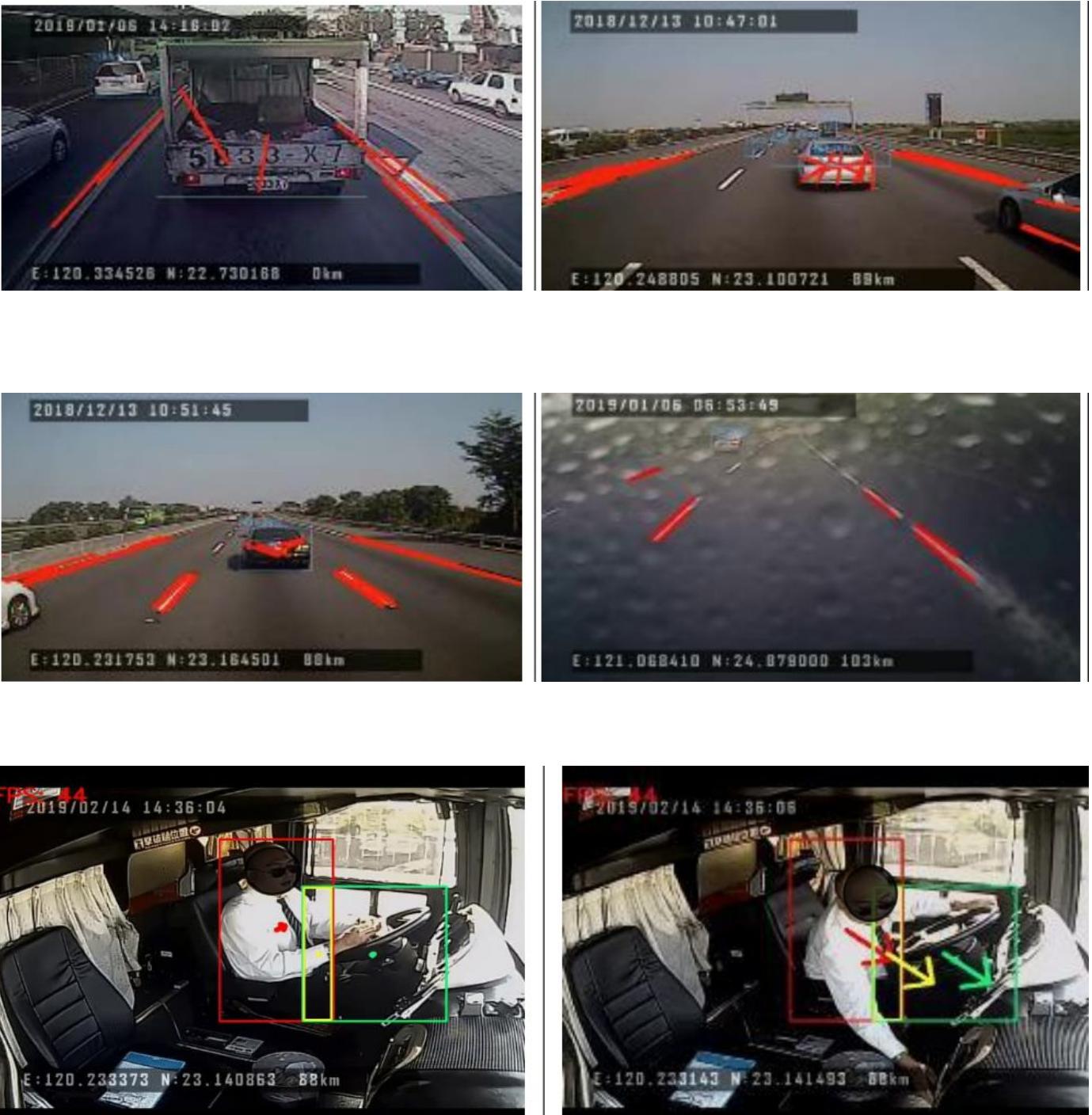
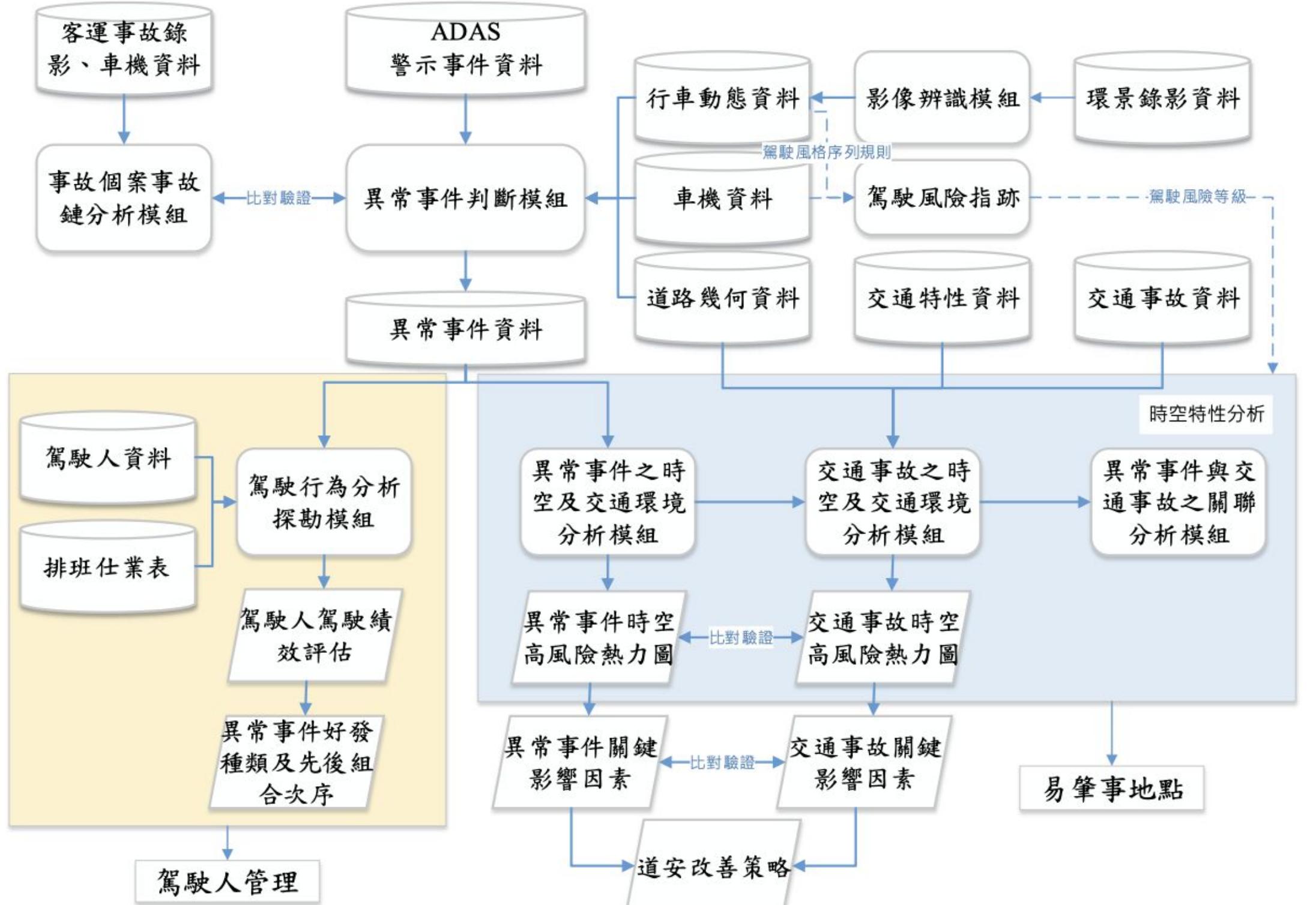


圖 1.4-1 整合模組架構

Inferring high-resolution traffic accident risk maps based on satellite imagery and GPS trajectories

Songtao He
MIT CSAIL
songtao@mit.edu

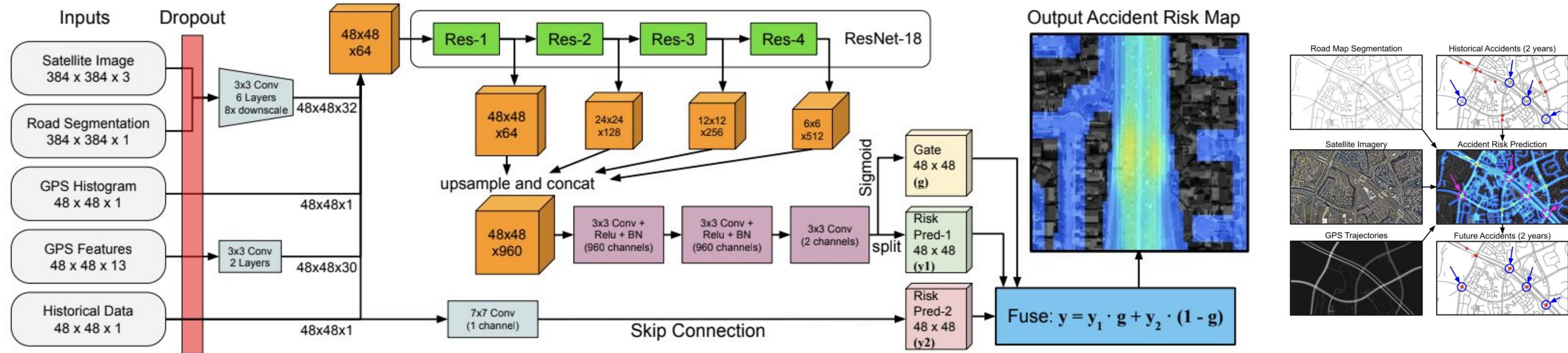
Mohammad Amin Sadeghi
HBKU QCRI
MSadeghi@hbku.edu.qa

Sanjay Chawla
HBKU QCRI
schawla@hbku.edu.qa

Mohammad Alizadeh
MIT CSAIL
alizadeh@csail.mit.edu

Hari Balakrishnan
MIT CSAIL
hari@csail.mit.edu

Samuel Madden
MIT CSAIL
madden@csail.mit.edu



易肇事路段 警示功能

Garmin特別整理台北市政府交通局肇事資料庫的易肇事路段警示資訊
貼心提醒，讓您安心駕駛！

行車請小心！台北市政府交通局根據往年死亡受傷人數和肇事件數等指標，加上現場會勘結果，整理出多處危險易肇事路段。像台北市幾個車流量龐大的路段及路口，都是大家爾孰能詳的恐怖路段，而連接三重、蘆洲、新莊與台北市士林地區的重要道路—中正路，以及聯繫中永和與台北市公館地區重要的十字路口—基隆路與羅斯福路交叉口，更是台北市十大交通易肇事路段與路口的榜首，因此Garmin特別貼心的幫大家將台北市政府交通局肇事資料庫公告的易肇事路段警示資訊，結合到Garmin導航機內，提供台北市49個危險路口提示，當Garmin的使用者行經易事路段時，機台會以「咚咚」聲及倒數距離顯示，來提醒駕駛人提高注意，遵守交通規定，不爭道、不搶快，並注意後方來車，保持前後的安全距離，維護自己與他人的行車安全，聰明提示，貼心提醒！

註：下載易肇事路段警示功能後，Garmin導航機在導航狀態下，也不會影響TTS語音播報功能，使用者仍可透過導航機精準的語音引導行進。

資料來源：台北市政府交通局肇事資料庫。



A top auto safety group tested 14 partial automated systems – only one passed



Image: General Motors

/ As driverless cars falter, partially automated systems like Tesla's Full Self-Driving and GM's Super Cruise are gaining more prominence. But they may be just as problematic.

By [Andrew J. Hawkins](#), transportation editor with 10+ years of experience who covers EVs, public transportation, and aviation. His work has appeared in The New York Daily News and City & State.

Mar 12, 2024, 11:54 PM GMT+8



115 Comments (115 New)



關於我們

<https://github.com/pedestrianomamori/pedestrianomamori.git>



工具和人



親友團