

新竹市第四十三屆中小學科學展覽會

作品說明書

科 別：生活與應用科學(一)

組 別：國中甲組

作品名稱：行不行！行人安全警示系統

關 鍵 詞：Yolov7、Amb82 mini、影像辨識

編 號：114JA-I003

作品名稱：行不行！行人安全警示系統

摘要

「台灣的『地獄』交通是觀光問題」，這是《CNN》的一篇報導標題，直指影響台灣交通—特別是行人路權極不安全。

步行環境不良及事故問題造成家長擔憂，即使學校再近也要汽機車接送，不願讓孩子走路上學，為了解決學生上下學安全問題，及降低行人步行發生事故，我們設計了這個「行人安全警示系統」，人行道會亮白燈，避免車輛夜晚視線不佳而未注意人行道，當車輛靠近人行道時，則會亮閃爍紅燈及廣播聲音來警示車輛，避免疲勞駕駛或應注意而未注意人行道及行人，也提醒行人注意後方有車輛靠近，避免被後方車輛撞擊，車輛一旦進入人行道，系統則會拍照上傳雲端並通知警方開罰，讓收到罰單的駕駛警覺，讓車輛習慣不隨意進入人行道，讓行人安全行。

壹、前言

一、研究動機

去年行人受傷的統計表，可以看出行人大多為成年或老人或是幼童受傷，但這些也是在人行道行走的大宗，因為國中小學生大多為家長接送，學校再近也要汽機車接送，不願讓孩子走路上學，但是即使是大人也無法躲過車輛闖入人行道，撞到造成的危害，雖然新車備有輔助系統，可偵測車輛死角有無車輛或行人，但舊車往往無此設備，也無相關法規去強制執行，造成行人不安全，除了駕駛人員應注意前方狀況，行人也需知道走在人行道時是否有後方來車闖入，而造成受傷，所以我們針對人行道進行相關的研究，設計出「行人安全警示系統」，警示車輛勿進入人行道，也讓行人知道有車輛靠近而警覺，守護行人在人行道行走的安全，這為我們研究的動機。



113年1月~11月全國行人件數各年齡層分布

排序	年齡族群	件數	死亡人數	受傷人數	死傷人數
1	成年人(25-64歲)	7,481	82	7,007	7,089
2	高齡者(65歲以上)	5,737	245	5,257	5,502
3	兒童(0-12歲)	1,089	2	1,049	1,051
4	年輕人(18-24歲)	975	3	936	939
5	少年(13-17歲)	655	0	615	615
6	不明	75	0	54	54
總計		16,012	332	14,918	15,250
產製日期：民國 114 年 2 月 12 日					
備註：不明係指案件中當事者資料無身份證號及出生日期。					

112 年、113 年 1-11 月查詢結果為初估值

交通部路政及道安司

相關網站連結：[168 交通安全入口網](#)[交通部統計查詢網](#)[警政署統計查詢網](#)

二、研究目的

- (一)、 問題解析
- (二)、 人行道相關法規
- (三)、 人行道問題及環境安全改善
- (四)、 標線型人行道-行人安全警示系統

三、文獻回顧

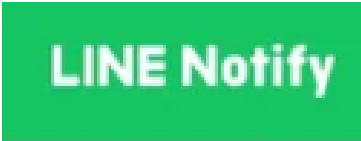
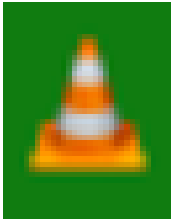
法蘭斯及夜市小霸王的 AMB 82 mini 講義。

貳、研究設備及器材

一、硬體設備及裝置

硬體	名稱
	無源蜂鳴器
	WS2812 燈條
	AMB 82-mini
	智慧型手機
	筆記型電腦

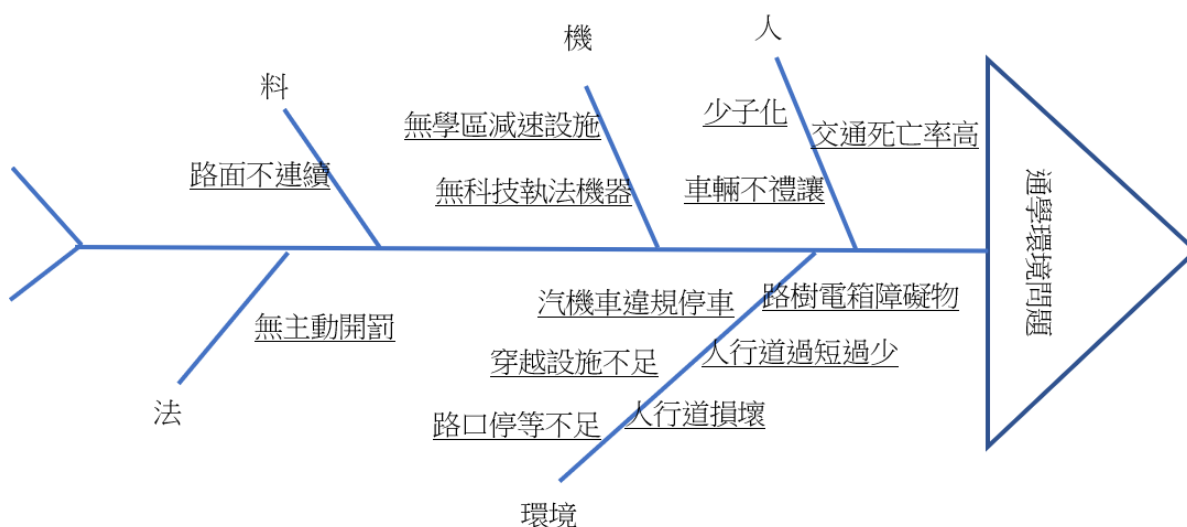
二、設計軟體及套件

軟體及網頁	名稱
	Arduino IDE
	BlocklyDuino F2
	Google Apps Script
	LINE Notify
	雲端硬碟
	VLC

參、研究過程或方法

目前步行環境不良及交通問題頻繁，造成家長擔憂，即使學校再近也要汽機車接送，不願讓孩子走路上學，造成大量排碳、違規停車、塞車…問題，為了解決這樣的問題，進行以下研究及改善。

一、問題分析



人行道有違停車婦沒得走 遭後方車撞傷脊椎

東森新聞
2023年11月12日



二、人行道相關法規

- (一)、 依據「道路交通管理處罰條例」(下稱處罰條例)第 3 條第 3 款規定，人行道指為專供行人通行之騎樓、走廊，及劃設供行人行走之地面道路…。」暨「道路交通標誌標線號誌設置規則」第 174 條之 3 規定「人行道標線，用以指示路面上僅限於行人行走之專用道。」標線型人行道與一般實體人行道無異。
- (二)、 另駕車違規行駛人行道者，違反處罰條例第 45 條第 1 項第 6 款規定，處駕駛人新臺幣（以下同）600 元以上 1,800 元以下罰鍰；違規臨時停車者，違

反同條例第 55 條第 1 項第 1 款規定，處駕駛人 300 元以上 600 元以下罰鍰；違規停車者，違反同條例第 56 條第 1 項第 1 款規定，處駕駛人 600 元以上 1,200 元以下罰鍰。至於在標線型人行道上堆積、置放障礙物或是未經許可擺設攤位，將開罰 1200~2400 元罰鍰。

三、人行道問題及環境安全改善

(一)、 改為人車分道：提升通學步道的安全及舒適性，並增加攝影檢舉，避免機車違停。

	改善前	改善後
路樹佔據	 「台南市仁愛國小通學步道優質化工程計畫」施工前，人行空間遭路樹佔據，民眾只能在車道上行走，步行時的安全性與路面平整質量極為堪憂。(內政部營建署提供)	 「台南市仁愛國小通學步道優質化工程計畫」施工後，校地內縮、打造圍籬增寬步道連結整個區域，活絡城市生命力。另外還增設家長接送區提供更優質的人行空間。(內政部營建署提供)
路面不平	 「高雄市前鎮區瑞豐國小武營路通學步道改善工程」施工前舊步道緣石高低參差不齊，步行環境較不安全。(內政部營建署提供)	 「高雄市前鎮區瑞豐國小武營路通學步道改善工程」施工後增添植栽、街燈、座椅等公共設施，以及建置無障礙系統，使視覺障礙者能在環境中安全步行。(內政部營建署提供)
路樹電箱佔據	 「臺中市霧峰區柳豐一街人行道工程」施工前無人行道配套措施的規劃、路樹生長茂密垂落至馬路上，景觀略顯雜亂。(內政部營建署提供)	 「臺中市霧峰區柳豐一街人行道工程」施工後社區層次分明，人車分道且展現其秩序性。(內政部營建署提供)



「漫步阿猴城-眷村周邊綠網串連計畫」施工前雜草叢生，周邊無適宜的徒步空間連接各歷史文化景點。(內政部營建署提供)

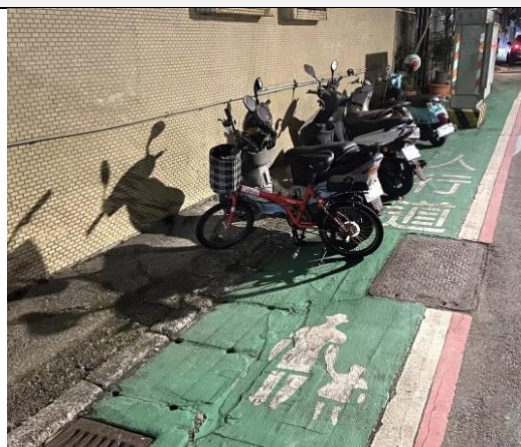


「漫步阿猴城-眷村周邊綠網串連計畫」施工後串聯周邊，具改善民眾休閒、觀光動線之效益，帶動屏東的地區發展。(內政部營建署提供)

(二)、 標線型人行道增設及改善：新增標線行人行道，並標示清楚避免汽車車占用或臨停，且使用「行人安全警示系統」保護行人安全。



改善前

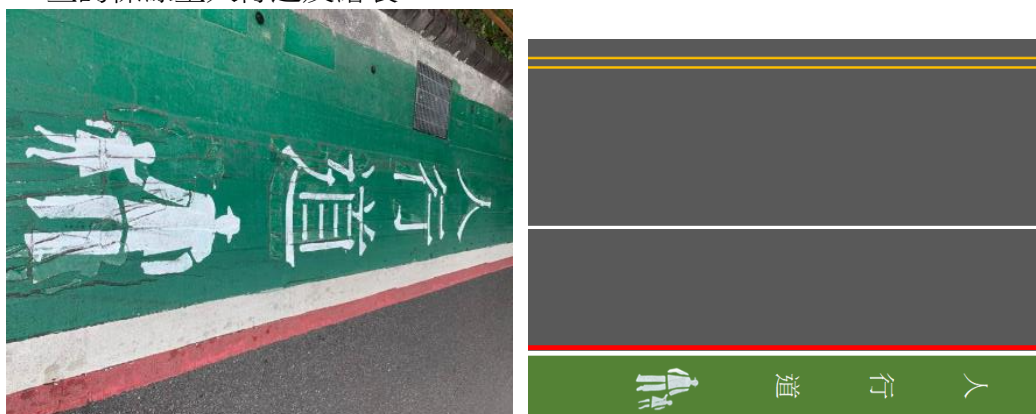


改善後

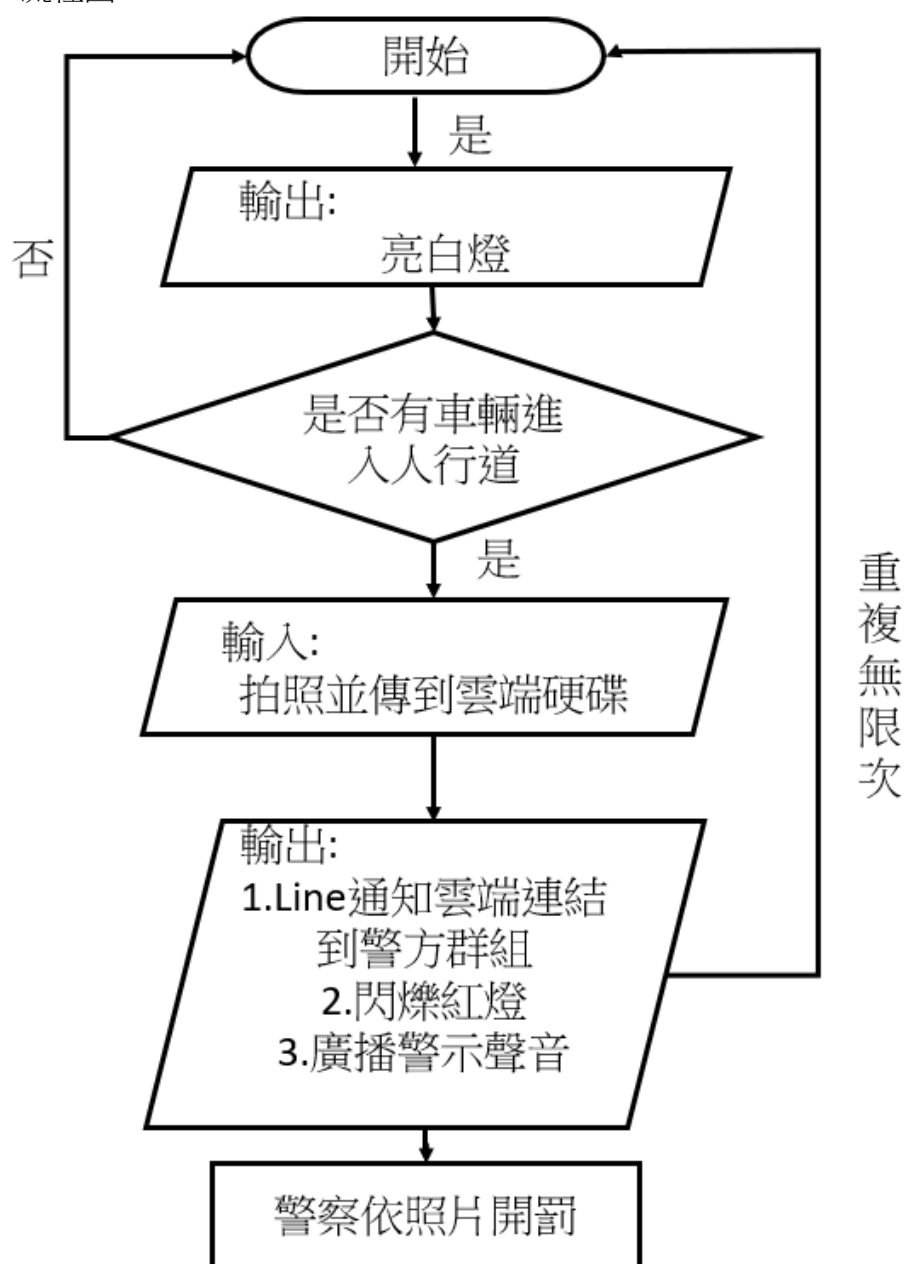


四、標線型人行道-行人安全警示系統-

(一)、查詢標線型人行道及繪製



(二)、流程圖



(三)、 AMB82 mini 程式

```
String appsScriptID =
"AKfycbw6Cr9o2wzyXi8_7Cj7bJnn9CmOCygsZWhLl_OgxBBx75FTDf71d_wjPv
SQrNM0jz-pYg";
String lineToken =
"Ar7tMH5b2ErX8UWgH446EuXpVjO2kOkLHEJVDDeW7EMN";
String folderName = "amb82-mini";
String fileName = "road";
int person = 0;
int cars = 0;
#include <WiFi.h>
WiFiSSLClient client;
#include "StreamIO.h"
#include "VideoStream.h"
#include "RTSP.h"
#include "NNObjectDetection.h"
#include "VideoStreamOverlay.h"
#define amb82_CHANNEL 0
#define CHANNELNN 3
#define NNWIDTH 576
#define NNHEIGHT 320
VideoSetting config(VIDEO_VGA, CAM_FPS, VIDEO_H264_JPEG, 1);
VideoSetting configNN(NNWIDTH, NNHEIGHT, 10, VIDEO_RGB, 0);
NNObjectDetection ObjDet;
RTSP rtsp;
StreamIO videoStreamer(1, 1);
StreamIO videoStreamerNN(1, 1);
int rtsp_portnum;
uint32_t img_addr = 0;
uint32_t img_len = 0;

#include "Base64.h"

char _lwifi_ssid[] = "WIFI";
char _lwifi_pass[] = "000000";
void initWiFi() {

    for (int i=0;i<2;i++) {
        WiFi.begin(_lwifi_ssid, _lwifi_pass);

        delay(1000);
        Serial.println("");
        Serial.print("Connecting to ");
        Serial.println(_lwifi_ssid);
```

```

    long int StartTime=millis();
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        if ((StartTime+5000) < millis()) break;
    }

    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("");
        Serial.println("STAIP address: ");
        Serial.println(WiFi.localIP());
        Serial.println("");

        break;
    }
}

String LineNotify(String token, String request) {
    String getAll="", getBody="";
    request.replace("%","%25");
    request.replace(" ","%20");
    //request.replace("&","%26");
    request.replace("#","%23");
    request.replace("\\","%22");
    request.replace("\n","%0D%0A");
    request.replace("%20stickerPackageId","&stickerPackageId");
    request.replace("%20stickerId","&stickerId");
    request.replace("%20imageFullsize","&imageFullsize");
    request.replace("%20imageThumbnail","&imageThumbnail");
    if (client.connect("notify-api.line.me", 443)) {
        client.println("POST /api/notify HTTP/1.1");
        client.println("Connection: close");
        client.println("Host: notify-api.line.me");
        client.println("User-Agent: ES8266/1.0");
        client.println("Authorization: Bearer " + token);
        client.println("Content-Type: application/x-www-form-urlencoded");
        client.println("Content-Length: " + String(request.length()));
        client.println();
        client.println(request);
        client.println();
        boolean state = false;
        long startTime = millis();
        while ((startTime + 3000) > millis()) {
            while (client.available()) {
                char c = client.read();

```



```

void MatrixLedAll(String color) {
    color.replace("#","");
    for(int i=0;i<(matrixString.length()/6);i++) {
        matrixString[i*6+0] = color[0];
        matrixString[i*6+1] = color[1];
        matrixString[i*6+2] = color[2];
        matrixString[i*6+3] = color[3];
        matrixString[i*6+4] = color[4];
        matrixString[i*6+5] = color[5];
        uint32_t R,G,B;
        R = (HextoRGB(color[0])*16+HextoRGB(color[1]));
        G = (HextoRGB(color[2])*16+HextoRGB(color[3]));
        B = (HextoRGB(color[4])*16+HextoRGB(color[5]));
        pixels.setPixelColor(i, pixels.Color(R, G, B));
    }
    pixels.show();
}

```

```

String HexReverse_s(int val, int pos) {
    int i = 0;
    String s = "0123456789abcdef";
    if (pos==1)
        i = (val-val%16)/16;
    else if (pos==2)
        i = val%16;
    return String(s[i]);
}

```

```

String SendStillToGoogleDrive(String myScript, String myFoldername, String
myFilename, String myLineType, String myLineToken, String myLineUserID,
String myImage, boolean capture) {

```

```

    const char* myDomain = "script.google.com";
    String getAll="", getBody = "";

```

```

    Serial.println("Connect to " + String(myDomain));
    if (client.connect(myDomain, 443)) {
        Serial.println("Connection successful");
        if (capture) {
            Camera.getImage(0, &img_addr, &img_len);
        }
        uint8_t *fbBuf = (uint8_t*)img_addr;
        size_t fbLen = img_len;

```

```

        char *input = (char *)fbBuf;

```

```

char output[base64_enc_len(3)];
String imageFile = "data:image/jpeg;base64,";
for (int i=0;i<fbLen;i++) {
    base64_encode(output, (input++), 3);
    if (i%3==0) imageFile += urlencode(String(output));
}
String Data =
myFoldername+myFilename+myLineType+myLineToken+myLineUserID+myImage;

```

```

client.println("POST " + myScript + " HTTP/1.1");
client.println("Host: " + String(myDomain));
client.println("Content-Length: " + String(Data.length()+imageFile.length()));
client.println("Content-Type: application/x-www-form-urlencoded");
client.println("Connection: close");
client.println();

```

```

client.print(Data);
int Index;
for (Index = 0; Index < imageFile.length(); Index = Index+1024) {
    client.print(imageFile.substring(Index, Index+1024));
}

```

```

int waitTime = 10000;
long startTime = millis();
boolean state = false;

```

```

while ((startTime + waitTime) > millis())
{
    Serial.print(".");
    delay(100);
    while (client.available())
    {
        char c = client.read();
        if (state==true) getBody += String(c);
        if (c == '\n')
        {
            if (getAll.length()==0) state=true;
            getAll = "";
        }
        else if (c != '\r')
            getAll += String(c);
        startTime = millis();
    }
    if (getBody.length()>0) break;
}

```

```

    }
    client.stop();
    Serial.println(getBody);
}
else {
    getBody="Connected to " + String(myDomain) + " failed.";
    Serial.println("Connected to " + String(myDomain) + " failed.");
}

return getBody;
}

String urlencode(String str) {
    const char *msg = str.c_str();
    const char *hex = "0123456789ABCDEF";
    String encodedMsg = "";
    while (*msg != '\0') {
        if (('a' <= *msg && *msg <= 'z') || ('A' <= *msg && *msg <= 'Z') || ('0' <=
*msg && *msg <= '9') || *msg == '-' || *msg == '_' || *msg == '.' || *msg == '~') {
            encodedMsg += *msg;
        } else {
            encodedMsg += '%';
            encodedMsg += hex[(unsigned char)*msg >> 4];
            encodedMsg += hex[*msg & 0xf];
        }
        msg++;
    }
    return encodedMsg;
}

#ifndef __OBJECTCLASSLIST_H__
#define __OBJECTCLASSLIST_H__
struct ObjectDetectionItem {
    uint8_t index;
    const char* objectName;
    uint8_t filter;
};

ObjectDetectionItem itemList[80] = {
    {0, "person", 1},
    {1, "bicycle", 1},
    {2, "car", 1},
    {3, "motorbike", 1},
    {4, "aeroplane", 1},
    {5, "bus", 1},
    {6, "train", 1},
    {7, "truck", 1},

```

{8, "boat", 1},
{9, "traffic light", 1},
{10, "fire hydrant", 1},
{11, "stop sign", 1},
{12, "parking meter", 1},
{13, "bench", 1},
{14, "bird", 1},
{15, "cat", 1},
{16, "dog", 1},
{17, "horse", 1},
{18, "sheep", 1},
{19, "cow", 1},
{20, "elephant", 1},
{21, "bear", 1},
{22, "zebra", 1},
{23, "giraffe", 1},
{24, "backpack", 1},
{25, "umbrella", 1},
{26, "handbag", 1},
{27, "tie", 1},
{28, "suitcase", 1},
{29, "frisbee", 1},
{30, "skis", 1},
{31, "snowboard", 1},
{32, "sports ball", 1},
{33, "kite", 1},
{34, "baseball bat", 1},
{35, "baseball glove", 1},
{36, "skateboard", 1},
{37, "surfboard", 1},
{38, "tennis racket", 1},
{39, "bottle", 1},
{40, "wine glass", 1},
{41, "cup", 1},
{42, "fork", 1},
{43, "knife", 1},
{44, "spoon", 1},
{45, "bowl", 1},
{46, "banana", 1},
{47, "apple", 1},
{48, "sandwich", 1},
{49, "orange", 1},
{50, "broccoli", 1},
{51, "carrot", 1},
{52, "hot dog", 1},

```

{53, "pizza",          1},
{54, "donut",          1},
{55, "cake",           1},
{56, "chair",          1},
{57, "sofa",           1},
{58, "pottedplant",    1},
{59, "bed",            1},
{60, "diningtable",    1},
{61, "toilet",         1},
{62, "tvmonitor",      1},
{63, "laptop",         1},
{64, "mouse",          1},
{65, "remote",         1},
{66, "keyboard",       1},
{67, "cell phone",     1},
{68, "microwave",      1},
{69, "oven",           1},
{70, "toaster",        1},
{71, "sink",           1},
{72, "refrigerator",   1},
{73, "book",           1},
{74, "clock",          1},
{75, "vase",           1},
{76, "scissors",       1},
{77, "teddy bear",     1},
{78, "hair dryer",     1},
{79, "toothbrush",     1},
};
#endif

void ODPostProcess(std::vector<ObjectDetectionResult> results) {
    uint16_t im_h = config.height();
    uint16_t im_w = config.width();
    OSD.createBitmap(amb82_CHANNEL);
    if (ObjDet.getResultCount() > 0) {
        for (int i = 0; i < ObjDet.getResultCount(); i++) {
            int obj_type = results[i].type();
            if (itemList[obj_type].filter) {
                ObjectDetectionResult item = results[i];
                int xmin = (int)(item.xMin() * im_w);
                int xmax = (int)(item.xMax() * im_w);
                int ymin = (int)(item.yMin() * im_h);
                int ymax = (int)(item.yMax() * im_h);
                //printf("Item %d %s:\t%d %d %d %d\n\r", i,
                itemList[obj_type].objectName, xmin, xmax, ymin, ymax);
            }
        }
    }
}

```

```

        OSD.drawRect(amb82_CHANNEL, xmin, ymin, xmax, ymax,
3, OSD_COLOR_WHITE);
        char text_str[20];
        snprintf(text_str, sizeof(text_str), "%s %d",
itemList[obj_type].objectName, item.score());
        OSD.drawText(amb82_CHANNEL, xmin, ymin -
OSD.getTextHeight(amb82_CHANNEL), text_str, OSD_COLOR_CYAN);
        Serial.println((String(String(itemList[obj_type].objectName))+String(",
")+String(item.score())+String(",")+String(xmin)+String(",
")+String(ymin)+String(",")+String(xmax)+String(",")+String(ymax)+String(",
")+String((xmax-xmin))+String(",")+String((ymax-ymin)))));
        if ((String(itemList[obj_type].objectName)=="person")) {
            person = person + 1;
        }
        if ((String(itemList[obj_type].objectName)=="car")) {
            cars = cars + 1;
        }
    }
}

Serial.println((String("物件數 :")+String(ObjDet.getResultCount())+String(",
人數 :")+String(person)+String(", 車數 :")+String(cars)));
if (cars > 0) {
    for (int count = 0; count < 3; count++) {
        analogWrite(4,200);
        delay(300);
        analogWrite(23,200);
        MatrixLedAll("ff0000");
        delay(500);
        strTemp_ = HexReverse_s(0, 1)+HexReverse_s(0, 2)+HexReverse_s(0,
1)+HexReverse_s(0, 2)+HexReverse_s(0, 1)+HexReverse_s(0, 2);
        MatrixLedAll(strTemp_);
        delay(100);
        analogWrite(4,0);
        delay(100);
    }
    SendStillToGoogleDrive(String("/macros/s/") + appsScriptID + String("/exec"),
String("&myFoldername=") + urlencode(folderName),
String("&myFilename=") + urlencode(fileName), String("&myLineType=") + "",
String("&myToken=") + urlencode(lineToken),
String("&myUserID=") + urlencode(""), "&myFile=", 1);
}
MatrixLedAll("ffffff");
person = 0;
cars = 0;

```

```

        OSD.update(amb82_CHANNEL);
    }
    void setup()
    {
        Serial.begin(115200);
        delay(10);
        initWiFi();

        pixels.begin();
        pixels.show();
        config.setBitrate(2 * 1024 * 1024);
        Camera.configVideoChannel(amb82_CHANNEL, config);
        Camera.configVideoChannel(CHANNELNN, configNN);
        Camera.videoInit();
        rtsp.configVideo(config);
        rtsp.begin();
        rtsp_portnum = rtsp.getPort();
        ObjDet.configVideo(configNN);
        ObjDet.setResultCallback(ODPostProcess);
        ObjDet.modelSelect(OBJECT_DETECTION, DEFAULT_YOLOV7TINY,
NA_MODEL, NA_MODEL);
        ObjDet.begin();
        videoStreamer.registerInput(Camera.getStream(amb82_CHANNEL));
        videoStreamer.registerOutput(rtsp);
        if (videoStreamer.begin() != 0) {
            Serial.println("StreamIO link start failed");
        }
        Camera.channelBegin(amb82_CHANNEL);
        videoStreamerNN.registerInput(Camera.getStream(CHANNELNN));
        videoStreamerNN.setStackSize();
        videoStreamerNN.setTaskPriority();
        videoStreamerNN.registerOutput(ObjDet);
        if (videoStreamerNN.begin() != 0) {
            Serial.println("StreamIO link start failed");
        }
        Camera.channelBegin(CHANNELNN);
        OSD.configVideo(amb82_CHANNEL, config);
        OSD.begin();

        LineNotify(lineToken, "message="+String((Ip2String(WiFi.localIP()))));
    }

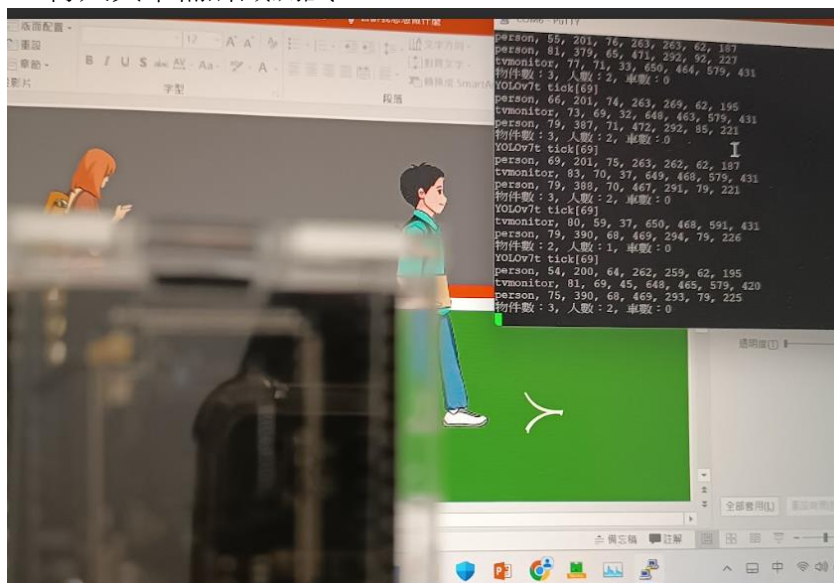
    void loop()
    {
    }

```

(四)、 Apps Script 撰寫，將照片上傳到雲端



(五)、 行人與車輛辨識測試



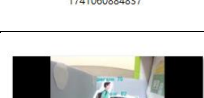
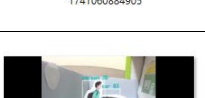
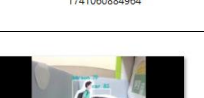
1. 車輛越近信賴指數越高

變因	紅車 辨識畫面 1	紅車 辨識畫面 2	紅車 辨識畫面 3	紅車 辨識畫面 4	信心 指數
車在 30 公分	 20250303222943-road	 20250303222931-road	 20250303222920-road	 20250303222907-road	車 65-69
車在 15 公分	 20250303225116-road	 20250303225105-road	 20250303225051-road	 20250303225038-road	車 89-91

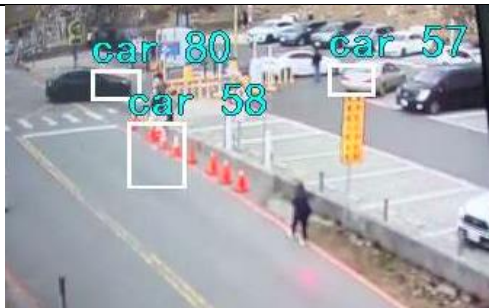
2. 車輛顏色越淺信賴指數越高

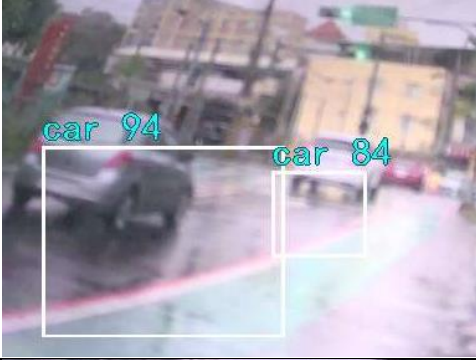
變因	車在 30 公分 辨識畫面 1	車在 30 公分 辨識畫面 2	車在 30 公分 辨識畫面 3	車在 30 公分 辨識畫面 4	信心 指數
紅車	 1741060886908	 1741060886939	 1741060886969	 1741060886999	紅車 72-88
藍車	 1741060886097	 1741060886127	 1741060886185	 1741060885883	藍車 54-89

3. 人車同時出現時，車輛越近信賴指數越高

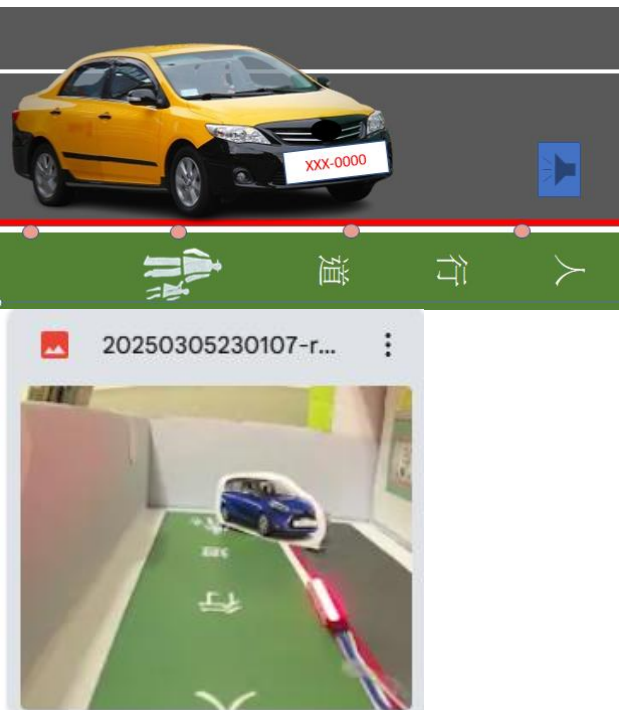
變因	人在 15 公分 辨識畫面 1	人在 15 公分 辨識畫面 2	人在 15 公分 辨識畫面 3	人在 15 公分 辨識畫面 4	信心 指數
車在 30 公分	 1741060884837	 1741060884905	 1741060884964	 1741060885052	車 52-82 人 73-81
車在 15 公分	 1741060884556	 1741060884634	 1741060884702	 1741060884770	車 82-83 人 78-79

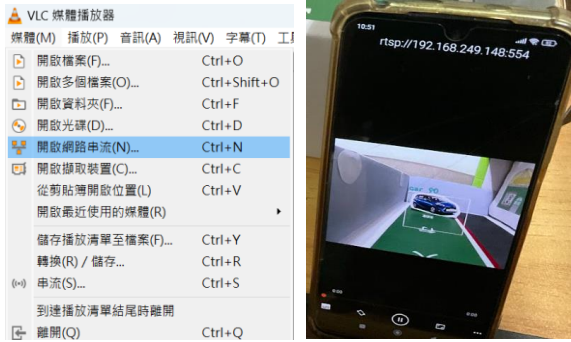
4. 道路測試

狀態	照片	信心指數
武嶺 鏡頭 君-車子 在動		車 58-80

武嶺 鏡頭 君-車 子靜 止			72
雨天 道路- 拍車 尾			84-94
雨天 道路- 拍車 頭			64-77
傍晚 超商 前面 機車 靜 止、 車輛 在動			56-65
傍晚 道路- 翻拍 VLC			91

(六)、完成測試

行人警示系統	說明
	標線人行道有白燈警示 車輛勿靠近
	當車輛進入人行道會閃亮 紅燈及聲音警示勿進入人 行道
	標線人行道有車輛或是 人進入，AMB82-mini 會 辨識
	AMB82-mini 辨識為車輛 則拍照上傳雲端

	Line notify 通知警方
	雲端照片供警方開罰使用
	警方也可以用 VLC 監看

肆、研究結果

一、問題解析

在既有的人行道改善成人車分道，讓行人免於車輛靠近而造成受傷，但空間不足的環境，只能靠標線型人行道來提供行人行走，行人暴露在沒有保護及護籬的狀態，需要額外的系統來保護行人安全。

二、人行道相關法規

目前雖然有相關罰則，但實際上標線型人行道之範疇遍布全臺各城市，警方對於此類違規較難執法，不乏出現汽機車違規行駛人行道之情形，所以設計「行人安全警示系統」時，減少警方奔波檢舉，直接將違規照片拍照及提供警方開罰這方面功能納入。

三、人行道問題及環境安全改善

人行道改善後及增加「行人安全警示系統」，讓行人不用隨時害怕後面突然的車輛，或是違停的車輛造成行走不便。

四、標線型人行道-行人安全警示系統

「行人安全警示系統」可以及時警示及拍照開罰及監控，在未有任何保護跟管理的標線型人行道，提供行人一個安全的保護。

伍、討論

- 一、了解目前人行道的法規，雖然法律有規定車輛闖入可以開罰，但往往各地的標線式人行道未統一，在店家門口的，往往汽機車會穿越去店家購買東西，這樣一來對行人也是危險，二來開罰時也無法開罰，應修改店家門口的標線型人行道，讓車輛不能穿越人行道。
- 二、系統辨識到車輛就拍照通知警方，可使用 line notify 及時通知警方，但此服務在2025年4月將取消服務，服務到期將改用其他方式來及時通知，若未及時通知警方，太慢開罰讓不在意，而達到的效果就會降低，未來改善成系統直接開罰單。

陸、結論

「行人安全警示系統」是有燈光聲音主動式提醒車主誤闖入標線型人行道，也提醒行人知道有車輛靠近，提高行人行走的安全，也利用辨識系統拍照，提供警方開罰，讓車主因罰單而自我管理，避免再次受罰，而不誤闖標線型人行道，讓行人脫離行人地獄，也讓家長可以放手讓孩子走路上學，行人安全行不行，可行。

柒、參考文獻資料

- 一、 Amb82mini 官方文件
<https://www.amebaiot.com/zh/ameba-arduino-summary/>
- 二、 Ameba IoT-AMB82 Mini 相關影片

https://www.youtube.com/results?search_query=Ameba+IoT

三、 泰山高中陳致中學習網站

<https://sites.google.com/tssh.ntpc.edu.tw/chenlung/ai/realtek-amb82-mini-ai-camera>

四、 法蘭斯 Amb82-mini 影像串流與影像辨識應用講義

五、 夜市小霸王 AMB 82 mini 講義