

中華民國第 62 屆中小學科學展覽會

作品說明書

高級中等學校組 電腦與資訊學科

052503

基於影像辨識之智慧冰箱學習系統

學校名稱：國立埔里高級工業職業學校

作者： 職三 李沛宸	指導老師： 鄭婉君
---------------	--------------

關鍵詞：影像辨識、增量學習、智慧冰箱

摘要

民眾在生活採買與準備食材中，經常面臨一些問題，例如：忘記冰箱食品而導致重複購買與浪費、食品放置過期...等。因此我們提出一個可以解決上述問題的系統，藉由影像辨識來判斷食品品項與移動軌跡，藉此建立冰箱內部的食品清單，並且可透過冰箱觸控螢幕與手機 APP，查看與設定清單內容。針對無法辨識的食品，系統可以學習訓練建立影像辨識模型，並針對現有的食品類別進行增量訓練，提高辨識食品的準確率。藉由本研究所提出的系統，可以學習與辨識各項冰箱食品、設定過期提醒通知，與冰箱異常偵測，協助使用者有效且便利地管理冰箱雜物、掌握冰箱的狀態。

壹、前言

在以往的智慧冰箱系統，若皆由使用者手動完成各項食材的建立與設定，整個過程耗時且麻煩，降低使用者使用意願。目前各大廠如 LG 與 Samsung 皆提出自家的智慧冰箱。LG InstaView 主打不用開門，只需要輕敲玻璃即可檢視冰箱內部的食品，減少開關冰箱門達到省電的目的，如圖 一。Samsung Family Hub 中 View Inside 的功能除了檢視冰箱食品外，可以輕觸冰箱畫面，並搭配 APP 建立食品品項名稱、保存期限等功能，如圖 二。然而對於冰箱內部繁雜的食品項目，皆依賴使用者手動輸入來建立食品清單。

本研究希望打造具備辨識與學習能力的智慧冰箱系統，可以為使用者自動化地建立與刪除冰箱食品清單。即便遇到無法辨識的食品，系統也能新增類別的功能，進行食品辨識模型的訓練。對於現有的食品類別，可藉由過去拿取與放入的食品影像，進行原有模型的增量訓練，提高對於食品的辨識精確度與可信度，提供使用者更好的體驗。



圖 一： LG InstaView [1]



圖 二： Samsung Family Hub [2]

一、 研究動機

在以往的智慧冰箱系統，在建立資料時都需要藉由 APP 來建立食物，經過拍照、輸入食食材品項、設定期限...等步驟，一一完成所有食材的建立，在每拿出冰箱時還需藉由 APP 來刪除食品資料，整個過程耗時且麻煩，長時間下來使用者的使用意願也會有所降低。因此想藉由此系統來改善，並且自動化建立與刪除食品資料的流程，達到使用者只須確認內容即可，在食品辨識模型的部分，也須隨著使用者的習慣，增加對於食品的辨識成功機率，讓使用者的體驗更加完善。

二、 目的

- (一) 開啟冰箱門時，系統會自動辨識食品品項，並且判斷食品的移動軌跡，建立食品清單。
- (二) 對於無法辨識的食品，系統可以增加食品類別，並建立辨識模型。
- (三) 利用每次辨識食品的影像，來對現有食品類別之模型進行增量訓練。
- (四) 可由冰箱的觸控螢幕與 APP，設定與更改食品清單內容之名稱、保存期限、提醒日期。
- (五) 偵測到冰箱的異常狀態時，以及針對已過期與即將過期的食品，系統會於冰箱觸控螢幕與 APP 顯示通知。

三、 文獻回顧

(一)智慧冰箱辨識食品技術

圖 三為各界對於智慧冰箱的研究結果，在早期辨識食品大部分是以 RFID、QR code 與重量感測等方式來偵測，在最近幾年影像辨識的技術越來越成熟，各項技術的介紹與比較如下：

Paper	Goal	Content recognition		Other hardware / software	User interaction	Availability
		Technology	Procedure			
[4]	platform for smart home design	RFID	automatic	housekeeping robot	not developed	tested only in a virtual environment
[5]	shopping list	RFID	automatic	smart oven	GUI	mainly sw simulated
[6]	food inventory, shopping list	RFID	automatic	microcontroller	web service interface	proof of concept
[7]	food inventory	RFID	automatic	-	smartphone app	pilot design
[8]	diet management	RFID	automatic	PC	smartphone app	testbed
[9]	food inventory, waste reduction	RFID	automatic	magnetic switch for open door	PC with touch screen	prototype
[10]	recipe suggestion, shopping list	barcode reader	user must scan the barcode	-	touch screen	computer simulated
[11]	food waste reduction	barcode, camera	user must scan, say or type the barcode	-	text or voice input, email if food expires	prototype: low identification rate
[12]	food inventory, shopping list	barcode reader, photodiodes	manual barcode scan and sensor calibration	microcontroller, temperature sensor	smartphone app	testbed
[13]	platform for smart home design	pressure sensor for egg holder	eggs are recognized only in their holder	light sensor for open door	email if eggs missing, alarm if door left open	case study for the platform evaluation
[14]	food inventory	weight sensor	user must say food name when entering it	screen	voice input	partial implementation of voice input methods
[15]	food inventory, shopping list	pressure sensor, photoresistor	only eggs, vegetables and drinks recognized	microcontroller	SMS if item missing, LED for notification	modules are tested separately
[16]	food status monitoring	no content recognition		gas, humidity and temperature sensors	email about food quantity and condition	testbed
[17]	food inventory	camera, force & proximity sensors	Google search by image in a browser	LED actuator to direct user attention	web service interface	testbed
[18]	monitoring of food quantity and status	camera, weight and gas sensors	image captured only, not recognized	microprocessor	smartphone app	testbed
[19]	food inventory and monitoring	camera	pictures analyzed by a server	temperature and humidity sensors	smartphone app	feasibility study
[20]	food waste reduction	inside and outside cameras	image captured only, not recognized	proximity and light sensors	touchscreen and smartphone app	testbed
[21]	food status monitoring	camera	only vegetables are classified	microprocessor	voice and SMS notification	algorithms of image analysis only
[22]	diet management	text scanner	manual data entering and scan of receipts	recommender system	natural language interaction	conversational assistant only
this paper	food inventory, waste reduction	RFID	automatic	microcontroller	natural language interaction	beta release ^a

圖 三：各界智慧冰箱的研究結果 [3]

1. RFID 原理介紹

RFID(Radio Frequency Identification)系統主要由 RFID Reader 與 RFID Tag 組成，RFID Tag 如圖 四， RFID Tag 有分成主動式、半被動式與被動式。因被動式的成本低、體積小，且不需電源，目前為市場上主流。能量來源是 RFID Reader 發射的無線電波，所以限制了讀取 RFID Tag 的範圍。



圖 四：RFID Tag[4]

2. Barcode

Barcode 為一維條碼，如圖 五，由不同寬度的黑白條紋，依不同編碼方式組合而成。利用掃描器接收由本身光源照射 Barcode 的反射光線，轉換成數位訊號經解碼、對比資料庫獲得條碼資訊。



圖 五：Barcode

3. 重量感測

如圖 六，利用感測器來得知目前冰箱內食品的重量與數量，此技術只適用於已規劃好位置的冰箱。



圖 六：重量感測器[5]

4. 辨識技術之比較

表 一為辨識技術的比較表，由此可知最適合作為食品的技术為影像辨識。重量感測器雖然功耗低，但食品只能放在特定位置。相較於 RFID 與 Barcode 要為每個食品加上讀取的標籤，影像辨識使用鏡頭限制低且便利性更高。

表 一：辨識技術比較表

辨識技術	RFID	Barcode	重量感測	影像辨識
設備需求	RFID Reader 與 Tag	Barcode Reader 與 Barcode	重量感測器	鏡頭
佔用空間	中	低	高	低
功耗	中	中	低	高
食品限制	每樣食品皆需 RFID Tag	每樣食品皆需 Barcode	食品須放置對應位置	無
便利性	中	中	低	高
準確性	依據使用者建檔完整性	依據使用者建檔完整性	僅測出食品總重	依據辨識模型判斷

(二)影像辨識平台與物件檢測工具

1. 影像辨識平台

目前影像辨識有 Create ML、AutoML、Teachable Machine、Lobe 等平台，來協助使用者訓練辨識的模型。表 二為各平台的比較，綜合下列比較的結果，Lobe 為免費平台且不需要連接網路。上述影像平台提供使用者便利的建立影像辨識模型，然而於此系統的情境下無法自動新增類別並訓練模型，所以本研究排除使用影像平台作為本系統影像辨識工具。

表 二：影像辨識平台比較

平台比較	Create ML	AutoML	Teachable Machine	Lobe
免費	是	否	是	是
使用限制	無	需要網路	需要網路	無
系統需求	Apple	不限	不限	Windows

2. 物件檢測(Object Detection)工具

物體檢測是使用邊界框(Bounding box)與已知物件類別來定義物件於影像中的位置。輸入資訊必須包含一個或多個物件的影像，經辨識後會輸出一個或多個邊界框，以及每個邊界框的類別標籤(Label)。

(1) R-CNN (Region Convolutional Neural Network)

使用影像中區域進行卷積神經網路(CNN)特徵擷取後，再評估類別和邊框，Fast R-CNN，Faster R-CNN 皆基於此架構。

(2) SSD (Single Shot Multibox Detector)

Faster R-CNN 準確度高，卻需要龐大的計算量，很難達到 Real-time 檢測。SSD 為 R-CNN 另一種思維做法，使用單個深度卷積網路檢測影像中目標，直接預測物體邊界框和種類。

(3) YOLO (You Only Look Once)

將影像作為輸入並直接預測多個物件的邊界框的位置和種類標籤，透過整張影像的特徵預測每個 Bounding box，並且計算屬於某類別的機率。

總和上述，R-CNN 準確但相較其他方法，資源消耗較為龐大且檢測速度較慢。SSD 辨識速度比 R-CNN 較快，但準確度卻比 R-CNN 要差。YOLO 系列比前兩者檢測更加快速，兼具兩種方法的優點，所以本研究使用 YOLO 作為物件檢測之工具。

貳、 研究設備及器材

一、 硬體設備

表 三為系統各部分所使用之硬體設備與規格。

表 三：硬體功能與規格表

	名稱	功能
冰箱	Raspberry Pi 4	冰箱資料處理與上傳
	觸控螢幕	查看食品清單與設定
	視訊鏡頭 1 (冰箱門外中央處)	擷取影像用於新增食品類別
	視訊鏡頭 2 (冰箱門內中央處)	擷取影像用於辨識食品與軌跡
	微動開關	偵測開關門
	DHT11	讀取溫度
Server	筆電：ASUS ROG G15 GA502IU	
	CPU	R7-4800HS
	記憶體	24GB
	顯示卡	GTX 1660Ti Max-Q
手機	OPPO R15	1. 辨識冰箱食品影像與軌跡 2. 訓練辨識模型 3. 建立資料庫 4. 接收冰箱與手機的 Request
無線分享器	RT-N12+Wireless-N300	通訊與傳輸

二、軟體

表 四為系統各部分所使用之軟體與套件。

表 四：軟體與套件表

Fridge	Server	APP
Microsoft Visual Studio	Microsoft Visual Studio	Android Studio
Python	YOLO	
OpenCV	SQL Server	
MediaPipe	Anaconda	
Adafruit_DHT	NVIDIA CUDA toolkit	
numpy	OpenCV	
RPi.GPIO		

參、 研究過程或方法

一、 研究架構圖

圖 七為本研究之研究架構圖，如圖所示。

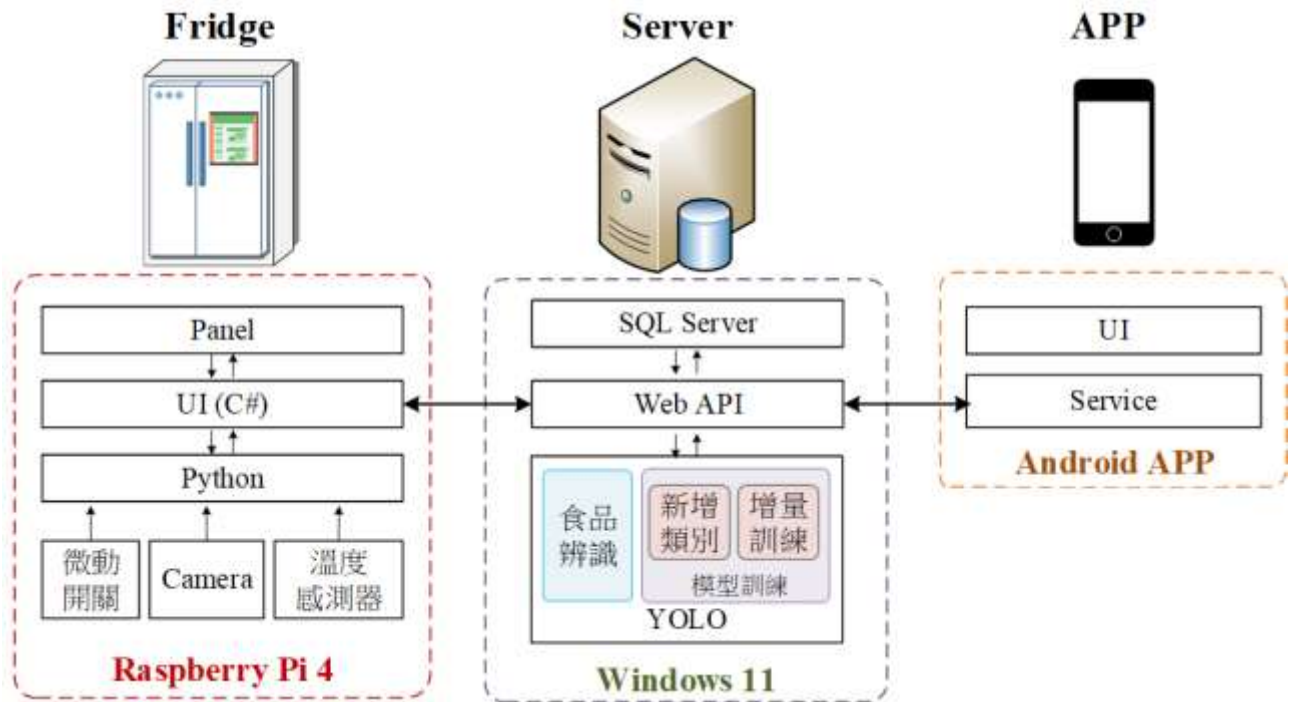


圖 七：系統架構圖

二、 研究流程與方向

冰箱使用樹莓派連接鏡頭與感測器，拍攝食品影像與收集感測資料傳送至 Server。樹莓派作業系統為 Raspberry Pi OS，並具備觸控式螢幕與使用者作互動。各部分的功能如圖 八，Server 負責食品辨識與模型訓練等功能，冰箱的觸控式螢幕與手機 APP 則為顯示與設定介面。

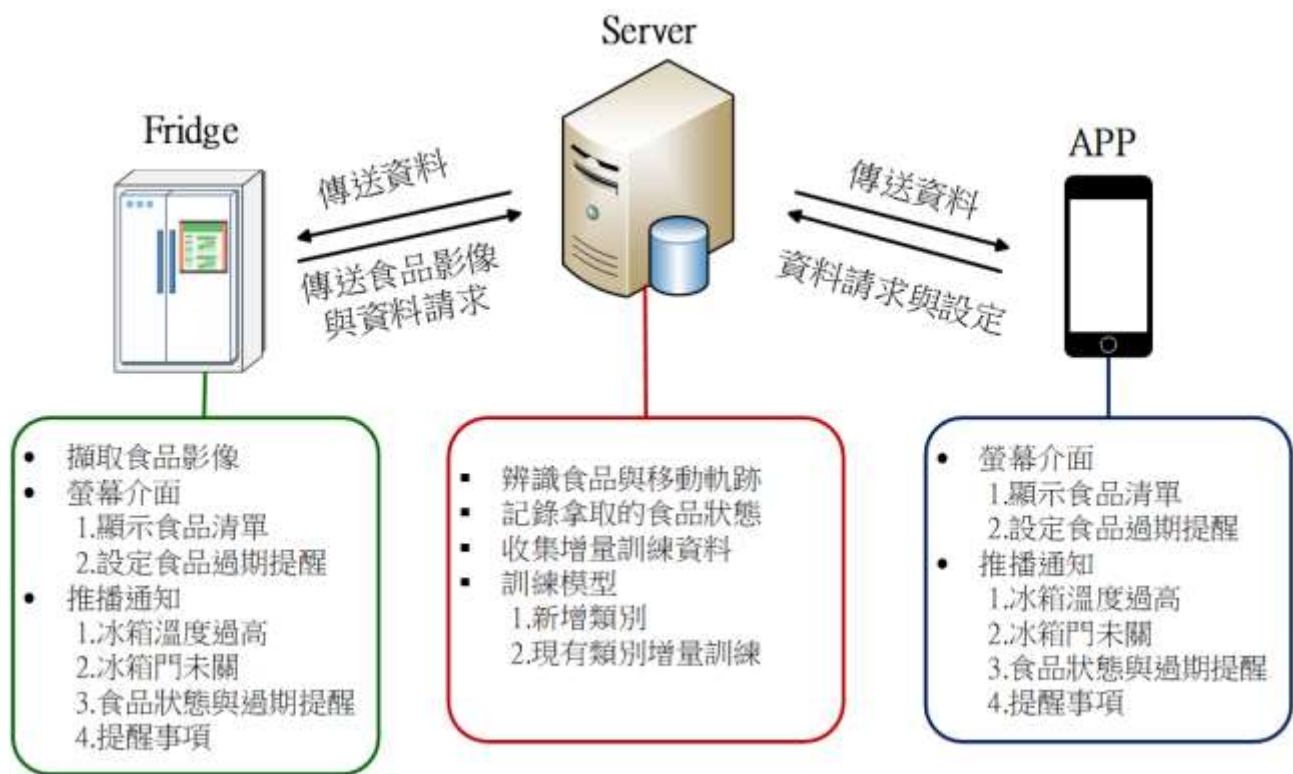


圖 八：系統功能表

三、 研究內容

系統研究依序由訓練食品模型以辨識不同的食品品項，使用者放入與拿取食品時，藉由影像辨識與軌跡判斷的資料，加以訓練模型，進一步提升辨識的準確率。系統偵測冰箱與食品狀態，傳送通知於冰箱介面與 APP。

(一)新增食品類別訓練模型

圖 九為新增食品類別訓練模型流程圖，依據功能可分為三個部分：1.物件偵測、2.設定 YOLO 參數、3.訓練模型。

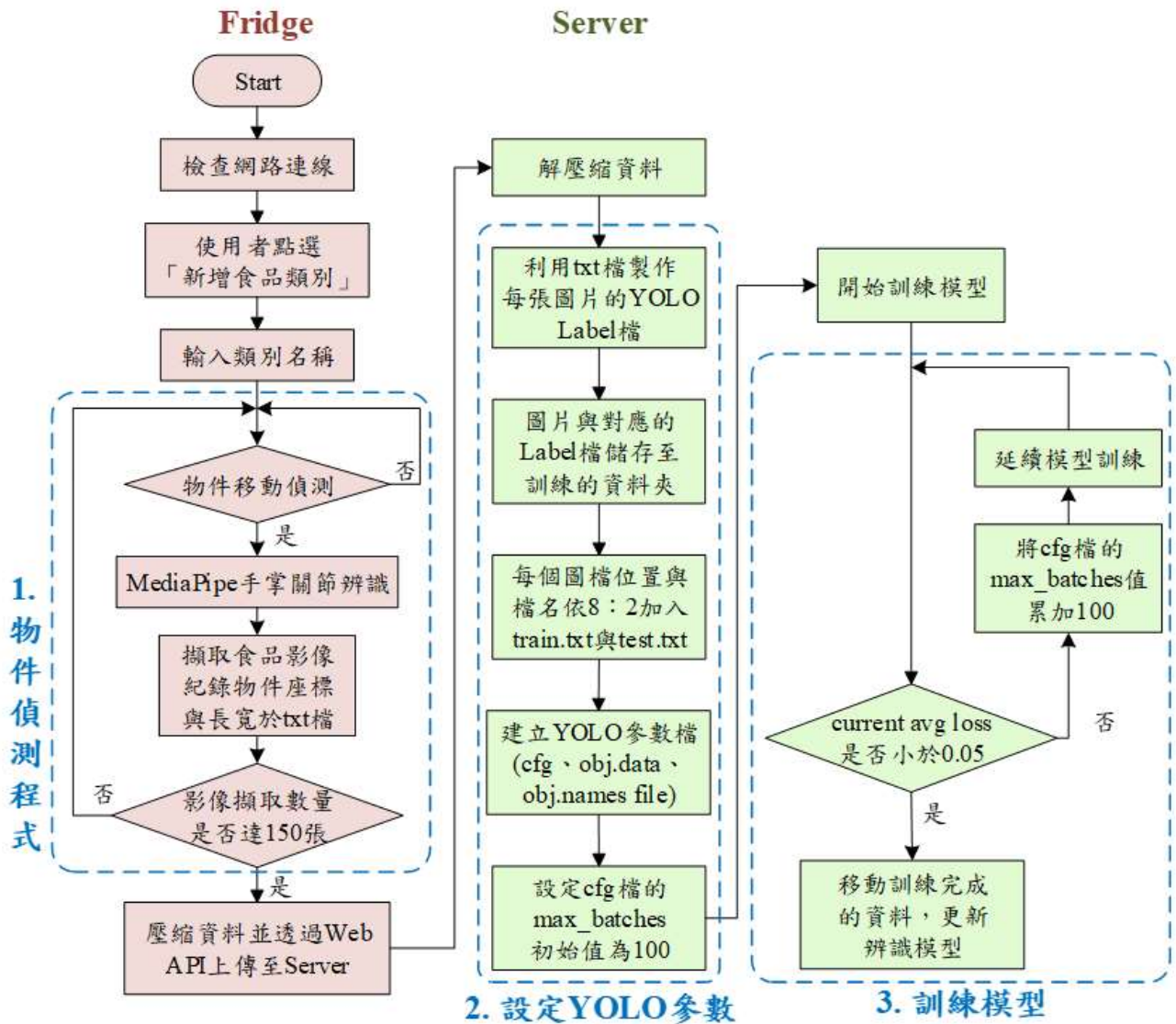


圖 九：新增食品類別與訓練模型流程圖

1. 物件偵測程式---移動偵測與手掌排除

使用者點選冰箱 UI 介面的「新增食品類別」，並且輸入完類別名稱後，執行以 Python 撰寫的「物件偵測程式」。程式開啟鏡頭且初始化設定後，比較每一影格與平均影像的差異，篩選變動程度大於門檻值(程式設定為 80)的區域，將變動面積大於設定值(程式設定為 600)的區域整合，形成圖 十、圖 十一中綠框的部分。

由於使用者在拍攝新增食品類別的照片時，系統要求使用者手持食品於冰箱前鏡頭轉動食品，以求能擷取各個角度的食品照片用來訓練模型。

「物件偵測程式」中使用 MediaPipe 套件以及訓練後的手部辨識模型，辨識畫面中手掌與各關節的位置，如圖 十、圖 十一黑框與描繪出的關節點。離左右邊界最遠的掌指關節為參考點，如下圖 十、圖 十一的藍色節點。

如圖 十、圖 十一所示，系統會將物件移動偵測所框選的區域，扣除參考點 X 軸以外的部分，為圖 十、圖 十一中紅框部分，作為辨識模型中所框選的物件。透過上述方法考量物件移動的區域與非食品物件的區域，能有效地框選適當的食品物件區域，提供更為精確的訓練模型資料，提高模型的辨識可信度與準確率。



圖 十：移動偵測與手掌排除-1



圖 十一：移動偵測與手掌排除-2

2. 設定 YOLO 參數

利用冰箱上傳的壓縮檔解壓縮，解壓縮後會有擷取的圖片和 txt 檔，txt 檔包含圖檔名稱座標位置與新增的品項名稱，依 txt 檔的內容來製作每個圖片的 YOLO 座標檔，如圖 十二，完成後儲存至訓練模型的資料夾，並且新加入的圖檔案位置與名稱依 8：2 的比例加入 YOLO 的 train.txt 與 test.txt，接著製作 YOLO 的 obj.data、obj.names 與 cfg 檔，將 cfg 檔的 max_batches 值設定初始值為 100。

3. 訓練模型

執行 YOLO 的程式訓練新的 weights 檔，執行完成便會檢查 current

avg loss 值是否低於 0.05，若未達成 max_batches 值會再增加 100 繼續執行，若達成則終止循環。下圖 十三為 weights 的訓練疊代次數收斂圖，其中經歷 9457 次疊代後，可得 current avg loss 值 0.2875，並且 mAP(Mean Average Precision)達到 97.4%。

```

while (!streamReader.EndOfStream)
{
    //獲取直到結尾
    string ram = streamReader.ReadLine(); //讀取一行
    string file_name = ram.Split(" ")[0]; //圖片名稱
    file_name_ram_before.Add(file_name);
    double[] xy = ram.Split(" ").Where((txt, index) => index != 0).Select(double.Parse).ToArray();
    //將陣列位置0以上的值轉換成浮點數
    StreamWriter streamWriter = new StreamWriter(extractPath+"\\ "+file_name+".txt", true);
    double x = (xy[0] + (xy[2] - xy[0]) / 2) * 1.0 / 1280;
    double y = (xy[1] + (xy[3] - xy[1]) / 2) * 1.0 / 960;
    double w = (xy[2] - xy[0]) * 1.0 / 1280;
    double b = (xy[3] - xy[1]) * 1.0 / 960;
    //轉換成YOLO格式
    streamWriter.Write(index + " " + x + " " + y + " " + w + " " + b);
    streamWriter.Close();
    streamWriter.Dispose();
    file_name = add_category + "_" + time + "_" + count;
    file_name_ram.Add(file_name);
    count++;
}

```

圖 十二：YOLO 參數轉換程式碼

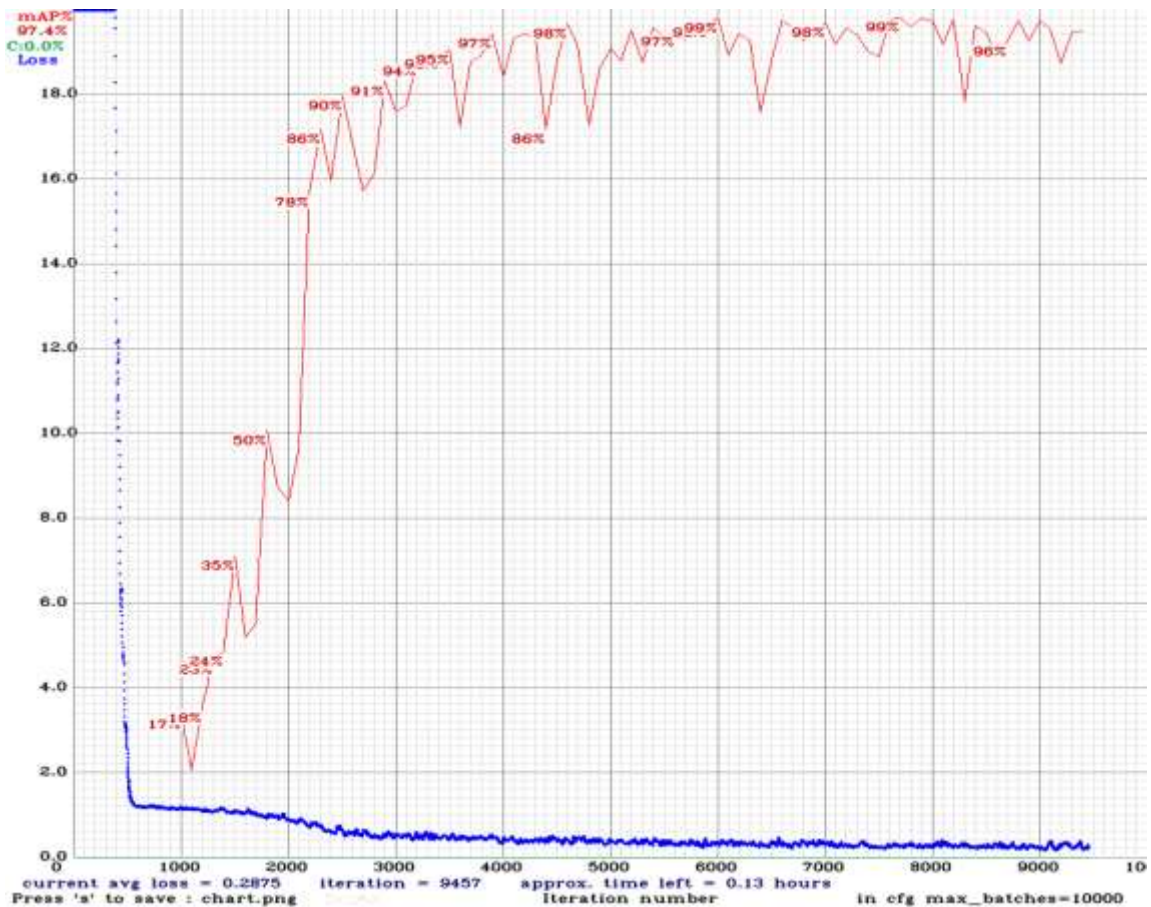


圖 十三：模型訓練準確率與 Average Loss Rate

(二)食品與動作辨識

圖 十四為放入與拿取食品辨識流程圖，當冰箱偵測到冰箱門開啟時，冰箱會擷取開門期間的影像，利用 Web API 上傳至 Server 進行辨識食品品項與食品的移動軌跡，並且分析判斷是否為放入冰箱，接續檢查食品清單或 buffer 食品清單中有沒有多個相同品項之食品，有則詢問使用者在更新資料庫，無則直接更新資料庫。

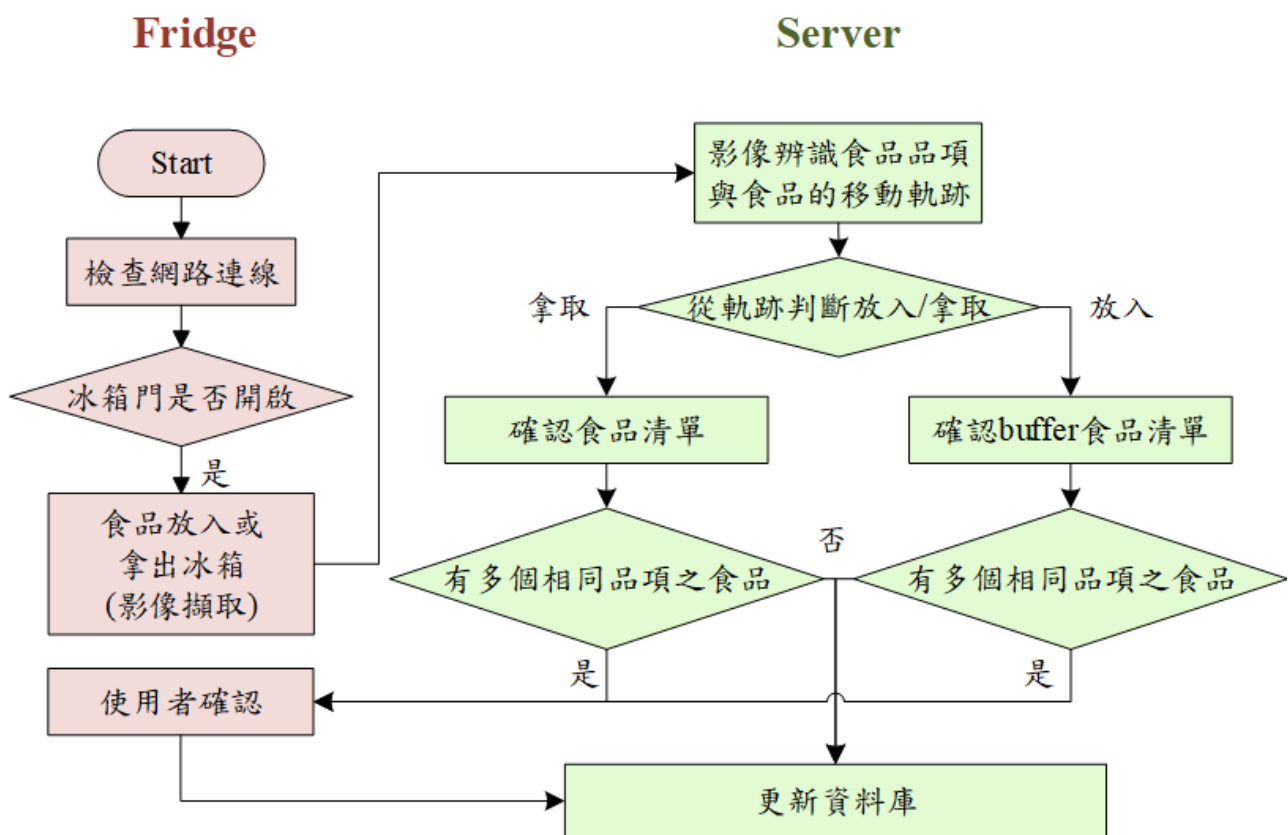


圖 十四：放入與拿取食品辨識流程圖

1. 食品品項辨識與軌跡判斷

使用者在拿取與放入食材時，冰箱將過程錄影下來透過 Web API 上傳至 Server 進行辨識，接收到檔案後執行 YOLO 的程式 darknet，如下圖 十五。辨識該上傳的影片，記錄下每影格可信度高於 50% 的辨識名稱、可信度與座標位置，再來分析座標的 x 軸變化將資料分片段依單趟拿取與放入

冰箱，在每段資料中依辨識的品項名稱加總可信度的值，該拿取或放入的食材則為加總最高值的品項名稱。

```
consol = Process.Start(cmd);
consol.StandardInput.WriteLine(@"cd C:\darknet-master\build\darknet\x64");//路徑移動到(C:\darknet-master\build\darknet\x64)
consol.StandardInput.WriteLine($"darknet detector demo {usePath}\obj.data {usePath}\123.cfg {usePath}\123_final.weights "
+ path + " -ext_output -dont_show -thresh .5");
//執行darknet程式，利用設定好的obj.data與123.cfg，還有訓練好的權重檔來辨識利用WebAPI上傳至Server的影片
consol.StandardInput.WriteLine("exit");
```

圖 十五：食品品項辨識與軌跡判斷程式碼

2. 拿取與放入

冰箱的 UI 介面程式剛開始執行時，會呼叫 python 的程式執行，並且由執行緒讀取 shell 顯示的值來對應處理，在開門時會執行建立一個 mp4 檔，並錄下影片，關門時會結束錄影，並透過讀取 shell 來得知檔案名稱並透過 Web API 上傳至 Server。

在辨識完食材名稱與分析出使用者的動作後，會依據放入與拿取分別執行不同的流程。若為拿取，則會搜尋食品資料表中是否有單個以上相同食品，有的話會詢問使用者是哪一筆資料，否則直接將資料移動到暫存食品資料表。若為放入，則會搜尋暫存食品資料表中是否有單個以上相同食品，有的話會詢問使用者是哪一筆資料，否則直接將資料移動到食品資料表，或者是直接新增一筆食品資料。在每辨識完影片，系統將會儲存該影片與每個品項名稱與座標位置，並且座標依單趟拿取與放入分別記錄。刪除與新增資料為以下圖 十六和圖 十七。

圖 十八、圖 十九辨識食品品項準確度與座標位置圖，圖 十八可看出訓練後的模型辨識照片中綠茶的準確度為 71%，並且顯示其座標與 Bounding box 長寬。

```

SqlCommand1 = new SqlCommand();
SqlCommand1.CommandText = "INSERT INTO food_inventory" +
    "(id,food_name,expiration_date,notification_date,owner,imagefilename,date)" +
    "VALUES(@id,@food_name,@expiration_date,@notification_date,@owner,@imagefilename,@date)";
//SQL新增資料的語法
SqlCommand1.Parameters.AddWithValue("@id", recordCount);
SqlCommand1.Parameters.AddWithValue("@food_name", food_name);
SqlCommand1.Parameters.AddWithValue("@expiration_date", expiration_date);
SqlCommand1.Parameters.AddWithValue("@notification_date", notification_date);
SqlCommand1.Parameters.AddWithValue("@owner", owner);
SqlCommand1.Parameters.AddWithValue("@imagefilename", imagefilename);
SqlCommand1.Parameters.AddWithValue("@date", date);
//加入值
SqlCommand1.Connection = sqlConnection;
SqlCommand1.ExecuteNonQuery();

```

圖 十六：加入食品資料表程式碼

```

SqlCommand1 = new SqlCommand();
SqlCommand1.CommandText = $"DELETE FROM food_inventory WHERE id={id}";
//SQL刪除資料的語法
SqlCommand1.Connection = sqlConnection;
SqlCommand1.ExecuteNonQuery();

```

圖 十七：拿出食品刪除資料程式碼

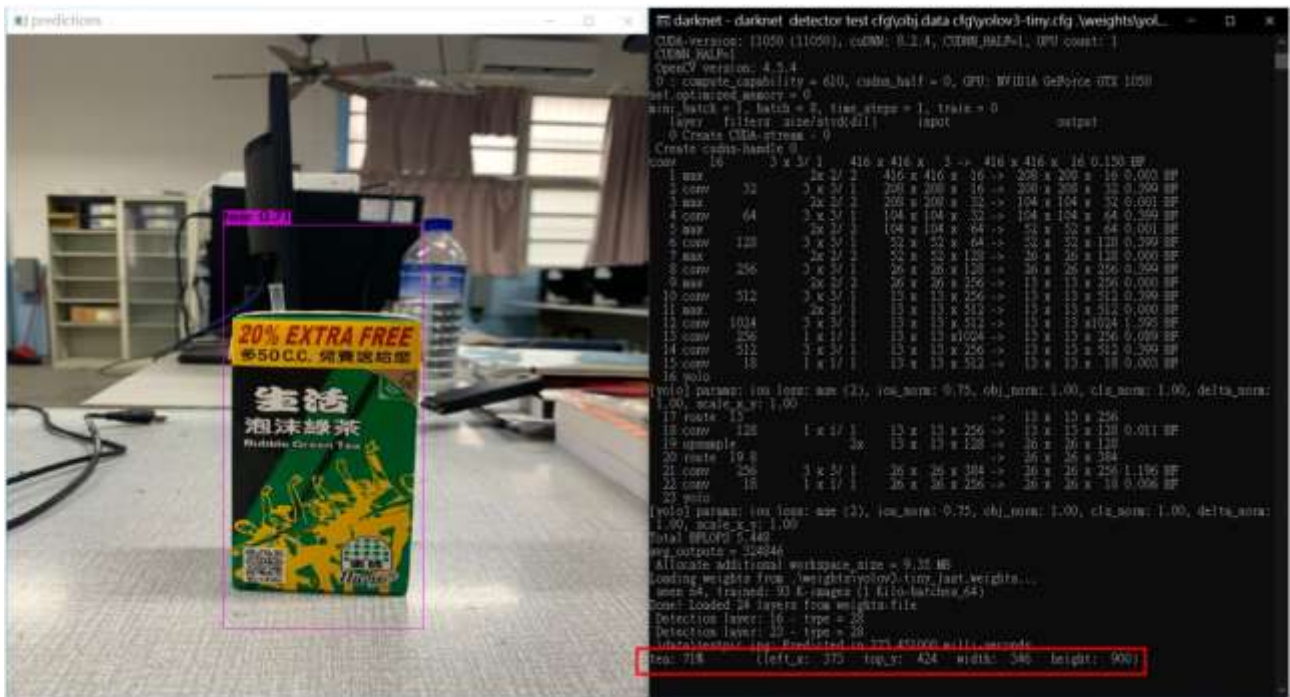


圖 十八：辨識食品類別準確度與座標位置-1

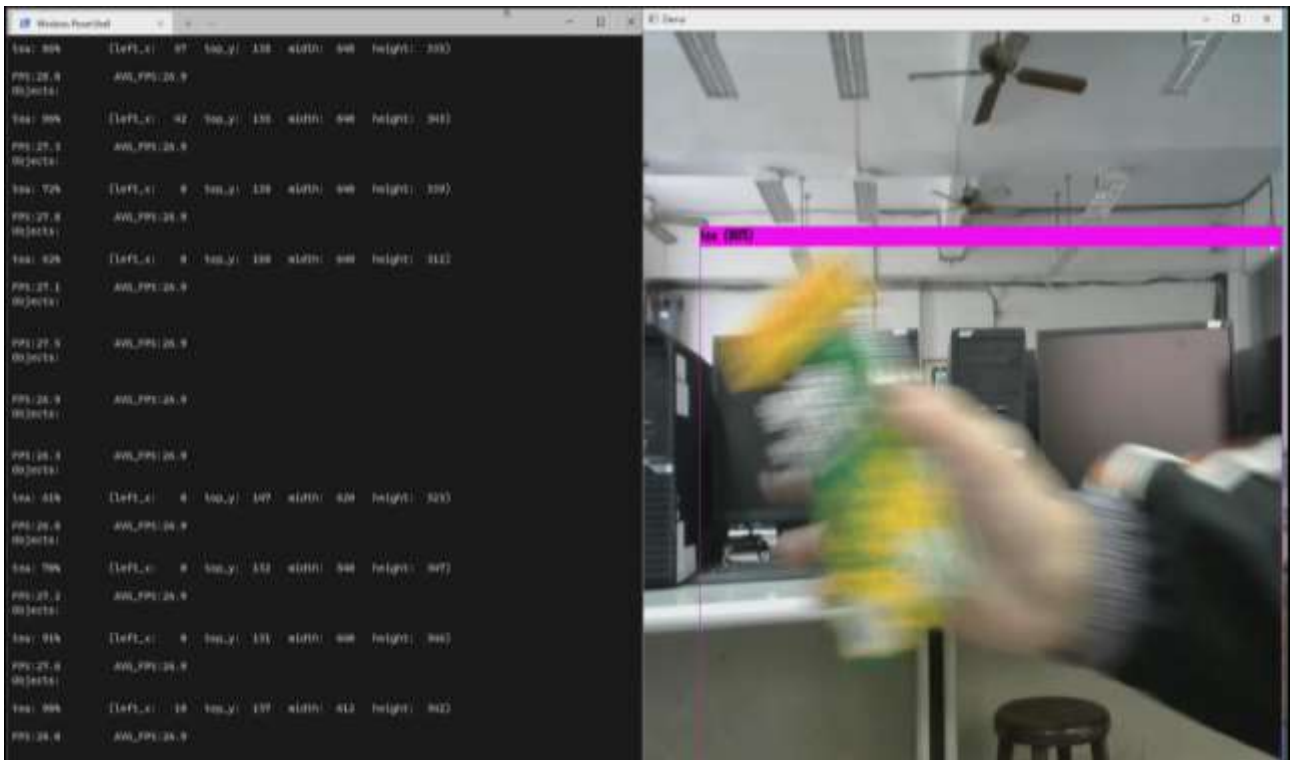


圖 十九：辨識食品類別準確度與座標位置-2

(三)既有類別增量訓練

圖 二十為既有類別增量訓練流程圖，依據食品放入/拿出的影像，作為增量訓練之資料，可分為三部分如下：

1. 收集增量訓練資料

將每次拿取與放入所辨識的影片，篩選每影格可信度高於 50%的辨識結果，將影片依品項與座標分別儲存與製作 YOLO Label 檔。

2. 設定 YOLO 參數

由新加入的圖片檔案位置依 8 比 2 的比例更新 YOLO 的 train.txt 與 test.txt，將 cfg 檔的 max_batches 值增加 100。

3. 訓練模型

執行 YOLO 的程式繼續訓練原本的 weights 檔，執行完成便會檢查 current avg loss 值是否低於 0.05，若未達成 max_batches 值會再增加 100 繼續執行，若達成則終止循環。圖 二十一為延續 weights 的模型訓練程式碼。

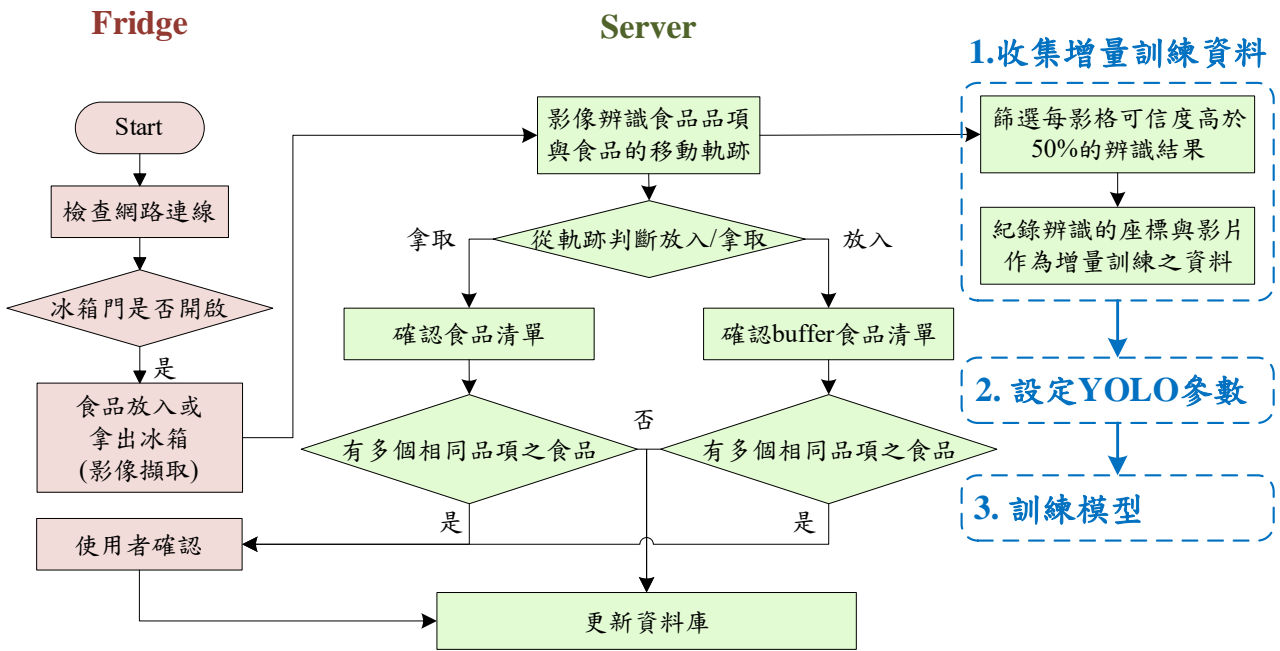


圖 二十：既有類別增量訓練流程圖

```

consol = Process.Start(cmd);
consol.StandardInput.WriteLine(@"cd C:\darknet-master\build\darknet\x64");
//路徑移動到(C:\darknet-master\build\darknet\x64)
consol.StandardInput.WriteLine($"darknet detector train {trainPath}\obj.data {trainPath}\\123.cfg " +
    $"C:\Users\paul0\Desktop\build_environment_train\Data\yolo-train\train\\123_final.weights");
//執行detector延續weights檔的訓練
consol.StandardInput.WriteLine("exit");

```

圖 二十一：既有類別增量訓練程式碼

(四)資料庫設計

表 五為資料表名稱、功能及欄位對照表。

表 五：資料庫設計表

資料表名稱	功能	欄位						
food_inventory	冰箱現有食品清單	id	食品名稱	保存期限	提醒日期	擁有者	圖片檔名	放入日期
		int	int	nvarchar	nvarchar	int	nvarchar	nvarchar
food_inventory_buffer	暫時拿出冰箱存放區	id	食品名稱	保存期限	提醒日期	擁有者	圖片檔名	放入日期
		int	int	nvarchar	nvarchar	int	nvarchar	nvarchar

food_class	食品類別名稱	<table border="1"> <tr> <td>id</td> <td>食品名稱</td> </tr> <tr> <td>int</td> <td>nvarchar</td> </tr> </table>	id	食品名稱	int	nvarchar				
id	食品名稱									
int	nvarchar									
owner	擁有者名稱	<table border="1"> <tr> <td>id</td> <td>擁有者</td> </tr> <tr> <td>int</td> <td>nvarchar</td> </tr> </table>	id	擁有者	int	nvarchar				
id	擁有者									
int	nvarchar									
notification	存放通知訊息	<table border="1"> <tr> <td>id</td> <td>標題</td> <td>內容</td> <td>圖片檔名</td> </tr> <tr> <td>int</td> <td>nvarchar</td> <td>nvarchar</td> <td>nvarchar</td> </tr> </table>	id	標題	內容	圖片檔名	int	nvarchar	nvarchar	nvarchar
id	標題	內容	圖片檔名							
int	nvarchar	nvarchar	nvarchar							
status_code	清單與通知的狀態值	<table border="1"> <tr> <td>清單狀態碼</td> <td>通知狀態碼</td> </tr> <tr> <td>nvarchar</td> <td>nvarchar</td> </tr> </table>	清單狀態碼	通知狀態碼	nvarchar	nvarchar				
清單狀態碼	通知狀態碼									
nvarchar	nvarchar									

(五)Web API 建置

表 六 Web API 功能參數對照表。

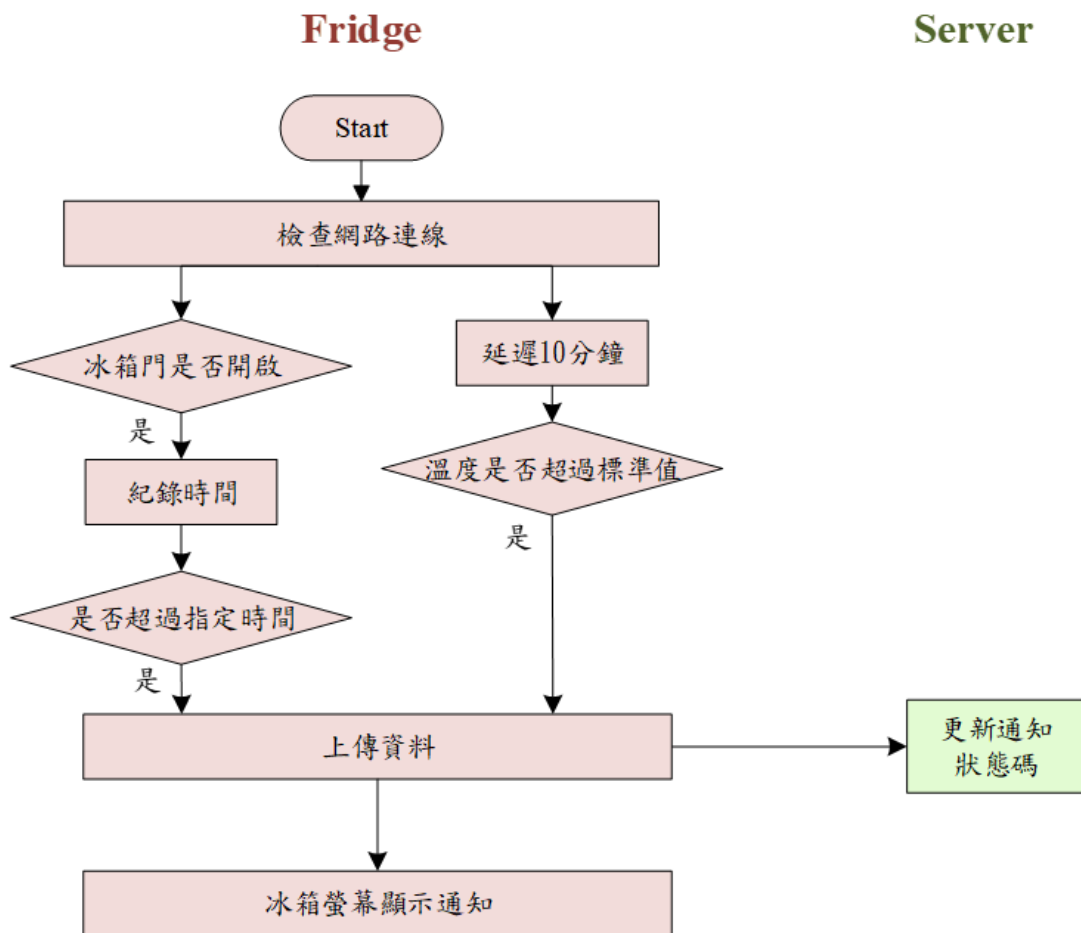
表 六：Web API 功能參數對照表

Web API	名稱	功能	參數	Body
Get	/Server/{filename}	請求圖片	圖片檔名	圖片
	/Server/database/{datasheet}/{select}/{id}	請求資料表特定資料	資料表名稱、選擇資料欄位與資料行id	JSON 格式的資料表的特定資料
Post	/fridge/TrainFile_Post	上傳訓練壓縮檔	-	-
	/fridge/FileUploads	上傳辨識影片	-	辨識結果
Put	/Server	更新資料表資料	JSON 格式的資料表、id、欄位與資料	-

(六) 冰箱狀態異常偵測

圖二十二為感測器異常通知流程圖，當冰箱門開啟時開始計時，若超過指定時間，冰箱之螢幕將發出警示訊息，並透過 Web API 更新通知狀態碼。

冰箱中的溫度傳感器則是每 10 分鐘讀取溫度，溫度範圍也會於冰箱螢幕顯示訊息與透過 Web API 更新通知狀態碼，以防冰箱中溫度過高超過正常值攝氏 7 度，導致食材的不新鮮。



圖二十二：冰箱狀態異常偵測流程圖

(七) 樹莓派與 APP 顯示介面與通知

如圖二十三，冰箱程式與 APP 在剛開始執行時，會讀取 Server 的食品清單內容，並且顯示於介面當中。Server 會讀取目前食品清單與通知的狀態碼，往後都會以每三秒來訪問 Server，在讀取到狀態碼的變動時，會依據狀態碼的

內容更新介面中的食品清單內容，再通知方面，也會依據狀態碼的內容向 Server 讀取通知的相關資料，並顯示通知提醒使用者。

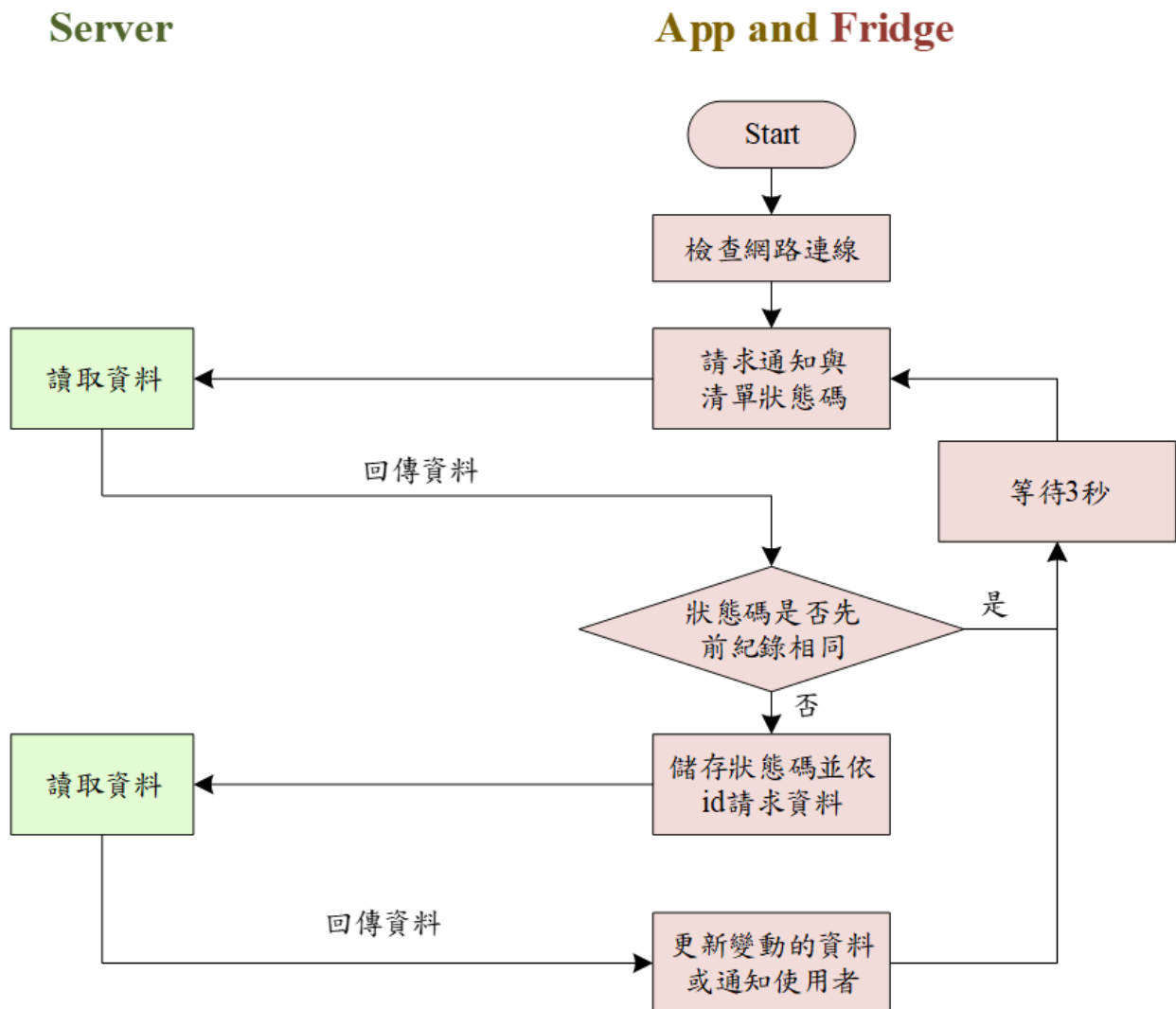


圖 二十三：樹莓派與 APP 顯示介面與通知流程圖

肆、 研究結果

一、 硬體基本建置

如圖 二十四為硬體基本建置：冰箱門後為 Camera 2，負責食品辨識與軌跡判斷。上方為與 Raspberry Pi 4 搭配之觸控螢幕，作為冰箱顯示介面。



Raspberry Pi 4
連接冰箱內部感測器與Camera，並顯示UI介面於觸控螢幕。



Server 辨識食品軌跡、紀錄食品清單並提供冰箱與APP與所需資料。
Camera 1 新增食品類別擷取影像。
Camera 2 辨識食品與移動軌跡。
溫溼度感測器 偵測冰箱內溫度。
微動開關 偵測冰箱門開關。
路由器 樹莓派、APP與Server連線之媒介。
觸控螢幕&手機 樹莓派與APP的顯示介面。

圖 二十四：硬體配置圖

二、 食品放入與拿出冰箱

(一)食品與軌跡辨識(辨識物體與動作加入清單)

圖 二十五、圖 二十六為使用者開啟冰箱門放入綠茶之流程，經由冰箱門內 Camera 2 擷取影片上傳至 Server 辨識，並依據 Server 之辨識結果更新食品清單，如圖 二十七為冰箱食品清單更新後之畫面。



圖 二十五：使用者開啟冰箱門



圖 二十六：使用者將食品放入冰箱

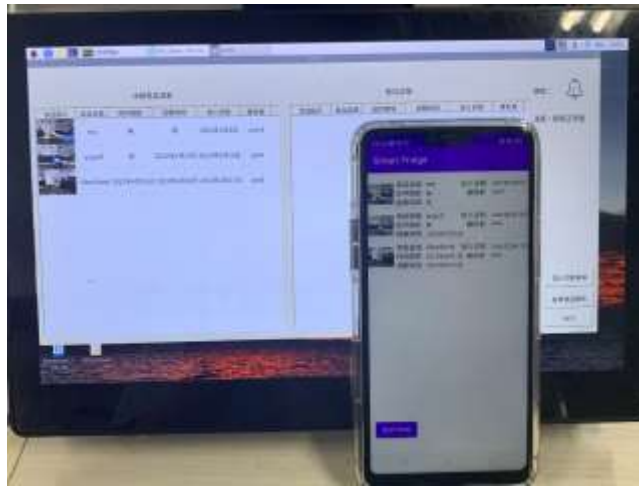


圖 二十七：冰箱食品清單更新的畫面

(二)食品確認與暫存清單

當冰箱內有相同品項的多個食品，使用者拿取其中一樣時，如圖 二十八、圖 二十九，系統會跳出辨識結果，並詢問使用者確切拿取的項目，經使用者確認後，此項目會被放置到食品暫存清單中，手機為食品清單顯示畫面，如圖 三十所示。



圖 二十八：使用者拿取其中一個優格



圖 二十九：使用者確認拿取的品項



圖 三十：食品清單與暫存清單畫面

(三)食品設定

點選食品項目會跳出食品設定的畫面，如圖 三十一所示，使用者可以設定食品名稱、保存期限、提醒日期與食品擁有者等功能。圖 三十二為手機食品設定之畫面。紀錄食品擁有者功能可適用於公司行號、學校宿舍...等公共區域。

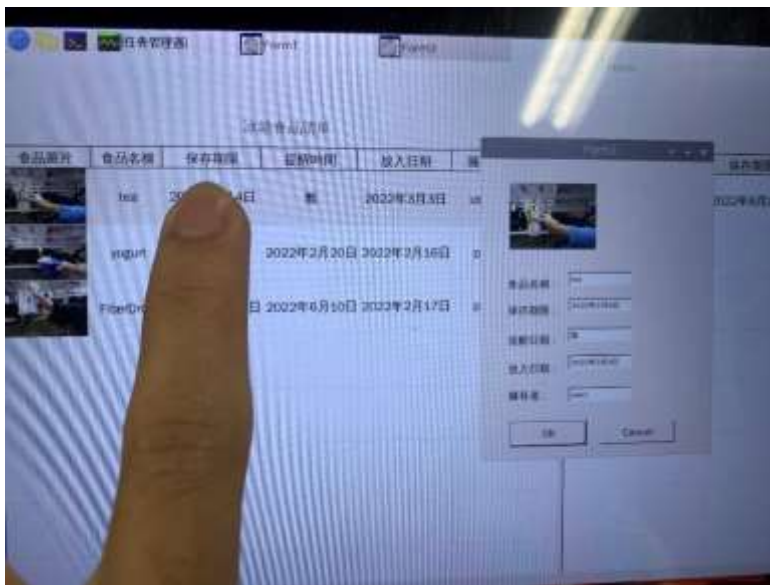


圖 三十一：食品設定畫面(冰箱介面)



圖 三十二：食品設定畫面(手機介面)

三、 新增食品類別

如圖 三十三、圖 三十四使用者點選冰箱介面中「冰箱食品類別」的按鈕，並且輸入食品名稱，系統會跳出「物件偵測程式」擷取 150 張食品影像，並上傳至 Server 進行模型訓練，訓練畫面如圖 三十五、圖 三十六所示。



圖 三十三：使用者點選新增類別按鈕

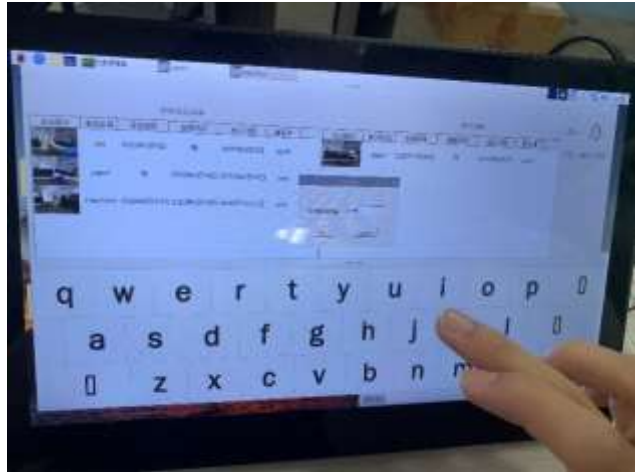


圖 三十四：使用者輸入食品類別名稱



圖 三十五：冰箱擷取預備訓練之圖檔



圖 三十六：Server 訓練模型之執行畫面

四、 冰箱異常與保存期限通知

系統偵測到冰箱異常與食品過期，會於 APP 與冰箱介面顯示通知，如圖 三十七、圖 三十七所示。



圖 三十七：APP 跳出冰箱異常通知畫面

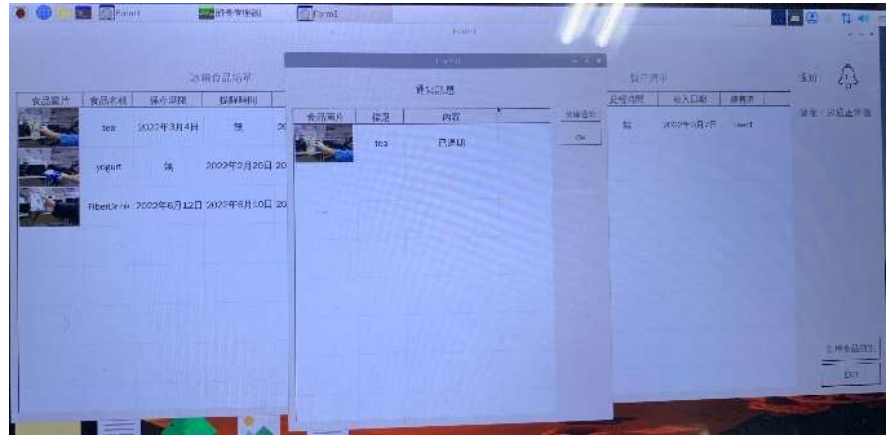


圖 三十八：冰箱跳出食品過期通知畫面

伍、 討論

表 七為本系統與市面上冰箱品牌的比較，在這之中小米與 LG 目前只有遠端查看即時的冰箱內部影像，三星是基於即時的冰箱內部影像，點選該食品達到紀錄食品的功能，本系統透過影像辨識建立食品清單，並透過學習訓練辨識新的食品品項，並且提供設定食品擁有者與提醒等功能。

表 七：各品牌智慧冰箱功能比較

功能 \ 廠牌	小米	LG	三星	本系統
影像辨識食品	×	×	×	✓
新增辨識食品類別	×	×	×	✓
紀錄食品與保存期限	×	×	✓	✓
遠端查看食品	✓	✓	✓	✓
紀錄食品擁有者	×	×	×	✓
提醒食品過期	×	×	✓	✓

陸、 結論

目前市面上推出許多智慧冰箱產品，但在辨識上有諸多限制，例如在遠端查看食品時，前方食品擋到後方的問題。本系統在建立個別食品資料的方面，利用影像辨識的技術，讓使用者不用一直輸入食品名稱，減化建檔步驟，讓使用體驗更流暢。

本系統可以改善一些食品放到過期、因不知道家中冰箱目前所存放的食品，而重複購買導致吃不完丟棄，從而降低被浪費的食材與無意義的花費，並且透過一次又一次的訓練模型，可以慢慢過濾出程式真正要辨識的食品，同時對於食品多樣化名稱也會有所紀錄。

未來可望進一步辨識食品的保存期限，提供更完善服務增進使用體驗。現代人飲食健康意識抬頭，然而因為生活忙碌，無形中忽略了飲食均衡，讓使用者的健康成為問題，因此在未來可以在本系統中增加對食品的營養分級，並於清單中顯示讓使用者參考。

本研究的系統未來可以結合分析使用者購買習慣與賣場特價資訊，當食品的存量低於設定時，Server 可以自動推薦冰箱存量過低的特價食品，並提供設自動訂購或外送平台之服務，為消費者節省更多時間與金錢。

本系統不僅可用於智慧冰箱之情境，也可用於其他需要物件辨識與建檔記錄等領域，如：工廠庫房、物流倉儲...等，皆可使用本研究作為系統基礎架構來延伸使用。

柒、 參考文獻資料

- [1] <https://www.lg.com/us/instaview-refrigerators>
- [2] <https://www.samsung.com/us/support/answer/ANS00049761/>
- [3] Renato Ferrero; Mohammad Ghazi Vakili; Edoardo Giusto, "Ubiquitous fridge with natural language interaction" 2019 IEEE International Conference on RFID Technology and Applications (RFID-TA)
- [4] <https://www.cool3c.com/article/150183>
- [5] https://shopee.tw/product/105691234/7137391275?utm_source=an_16136160000&utm_medium=affiliates&utm_campaign=-&utm_content=7435347a3958304269336f70473455743977394c----&af_siteid=an_16136160000&pid=affiliates&af_click_lookback=7d&af_viewthrough_lookback=1d&is_retargeting=true&af_reengagement_window=7d&af_sub_siteid=7435347a3958304269336f70473455743977394c----&c=-
- [6] <https://www.inside.com.tw/article/22381-not-so-smart-home-devices>

【評語】 052503

本作品以樹莓 Pi 開發板實作智慧冰箱部分功能，主要目的是透過影像辨識簡化冰箱內容物記錄的程序。雖然作品說明書最後有提供物品辨識、定位等的說明，但未提供完整的實驗規劃及實驗結果用以說明研究步驟每一個環節的可靠性（準確性）。此外，雖然此智慧冰箱的功能多且實作完整，但很多功能已經在之前得獎作品中出現，新增的功能在此延續性作品中並不多且佔整個作品的比重不大。

作品簡報

基於影像辨識之智慧冰箱學習系統

組別：高級中等學校組

科別：電腦與資訊學科

研究動機

民眾在生活採買與準備食材中，經常面臨一些問題，例如：忘記冰箱食品而導致重複購買與浪費、食品放置過期...等。以往的智慧冰箱系統，針對冰箱內部繁雜的食品項目，皆由使用者手動完成各項食材的建立與設定，整個過程耗時且繁瑣。



圖 1：LG InstaView

因此我們希望提出一個具備辨識與學習能力的智慧冰箱，藉由影像辨識自動化地建立冰箱內部的食品清單，並可透過冰箱觸控螢幕與 APP，查看與設定食品清單內容與狀態。



圖 2：Samsung Family Hub

研究目的

1. 開啟冰箱門時，系統會自動辨識食品品項與移動軌跡，建立食品清單。
2. 對於無法辨識的食品，系統可以新增食品類別，並建立辨識模型。
3. 利用每次辨識食品的影像，來對辨識模型進行增量訓練。
4. 可由冰箱的觸控螢幕與APP，設定與更改食品清單內容之名稱、保存期限、提醒日期。
5. 偵測到冰箱異常狀態，以及針對已過期與即將過期的食品，系統會於冰箱觸控螢幕與APP顯示通知。

系統功能與架構圖

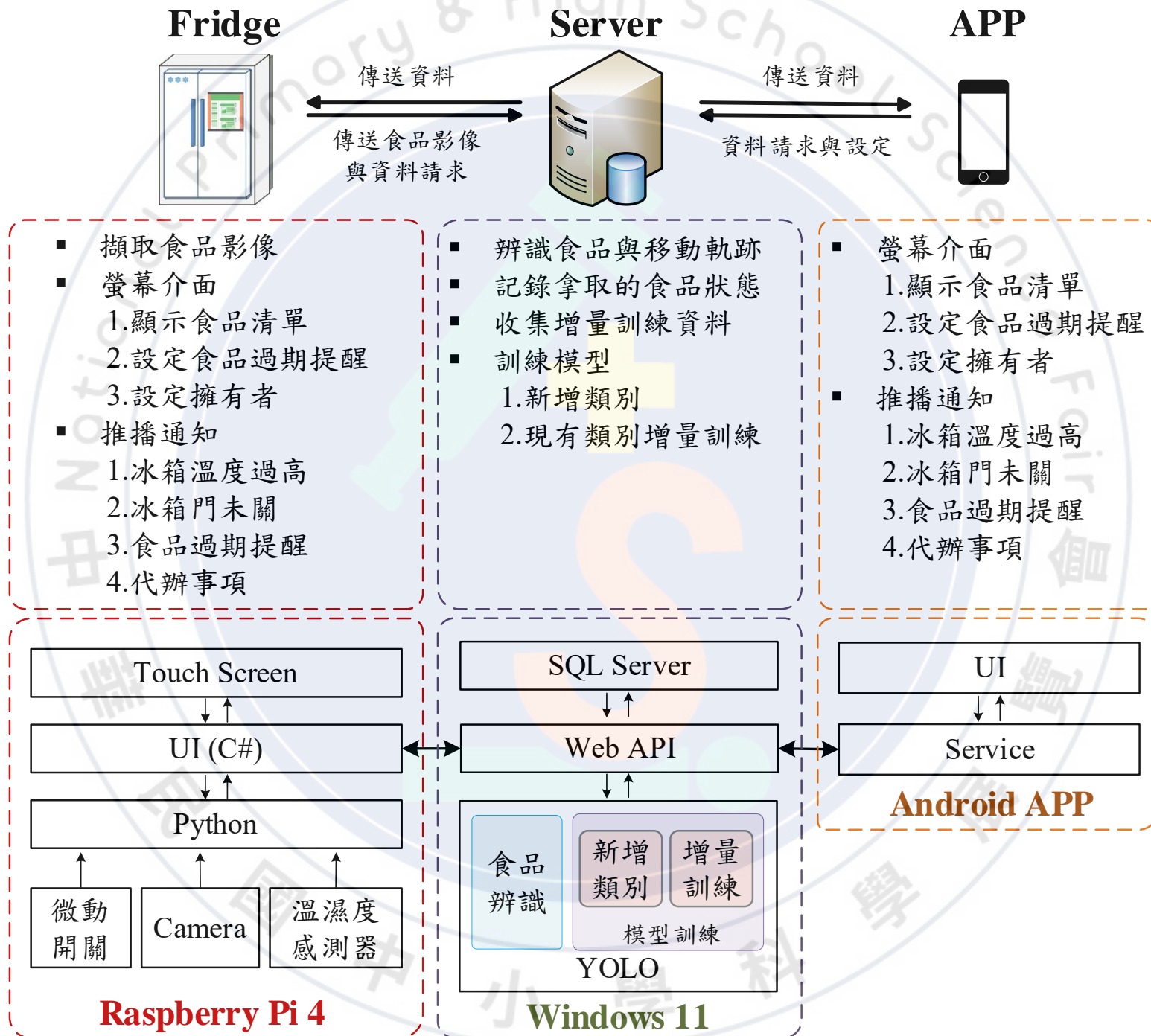
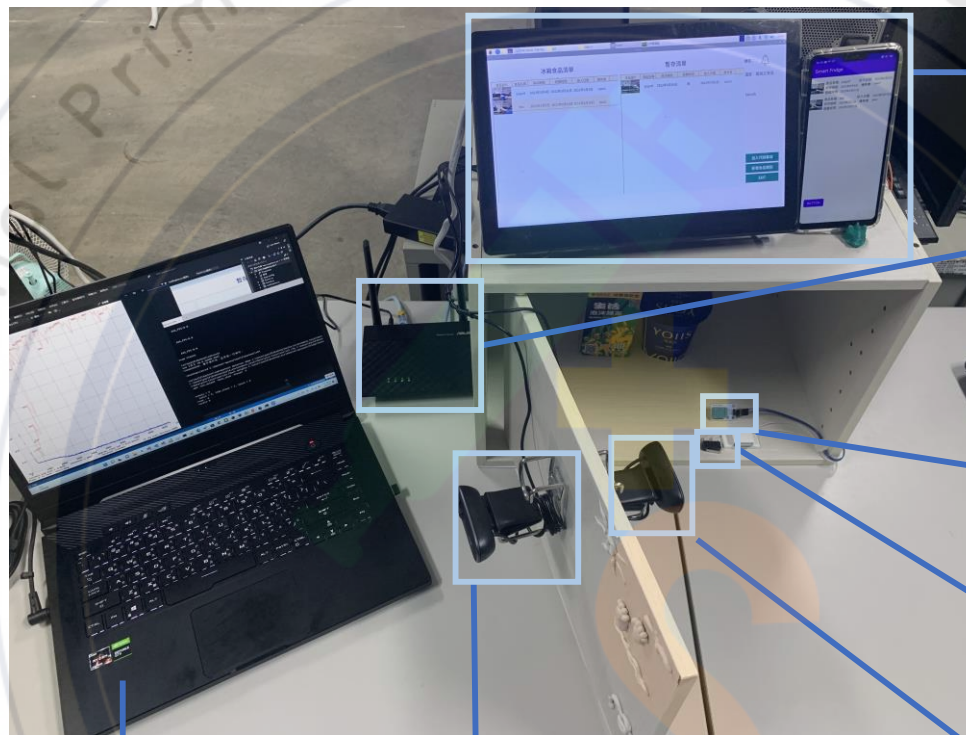


圖3：系統架構圖

硬體設備



Raspberry Pi 4
連接冰箱內部感測器與Camera，並顯示UI介面於觸控螢幕。



Server

辨識食品軌跡、紀錄食品清單並提供冰箱與APP與所需資料。

Camera 1

新增食品類別擷取影像。

觸控螢幕&手機

樹莓派與APP的顯示介面。

路由器

樹莓派、APP與Server連線之媒介。

溫溼度感測器

偵測冰箱內溫度。

微動開關

偵測冰箱門開關。

Camera 2

辨識食品與移動軌跡。

圖4：硬體配置圖

研究方法-新增食品類別(1/3)

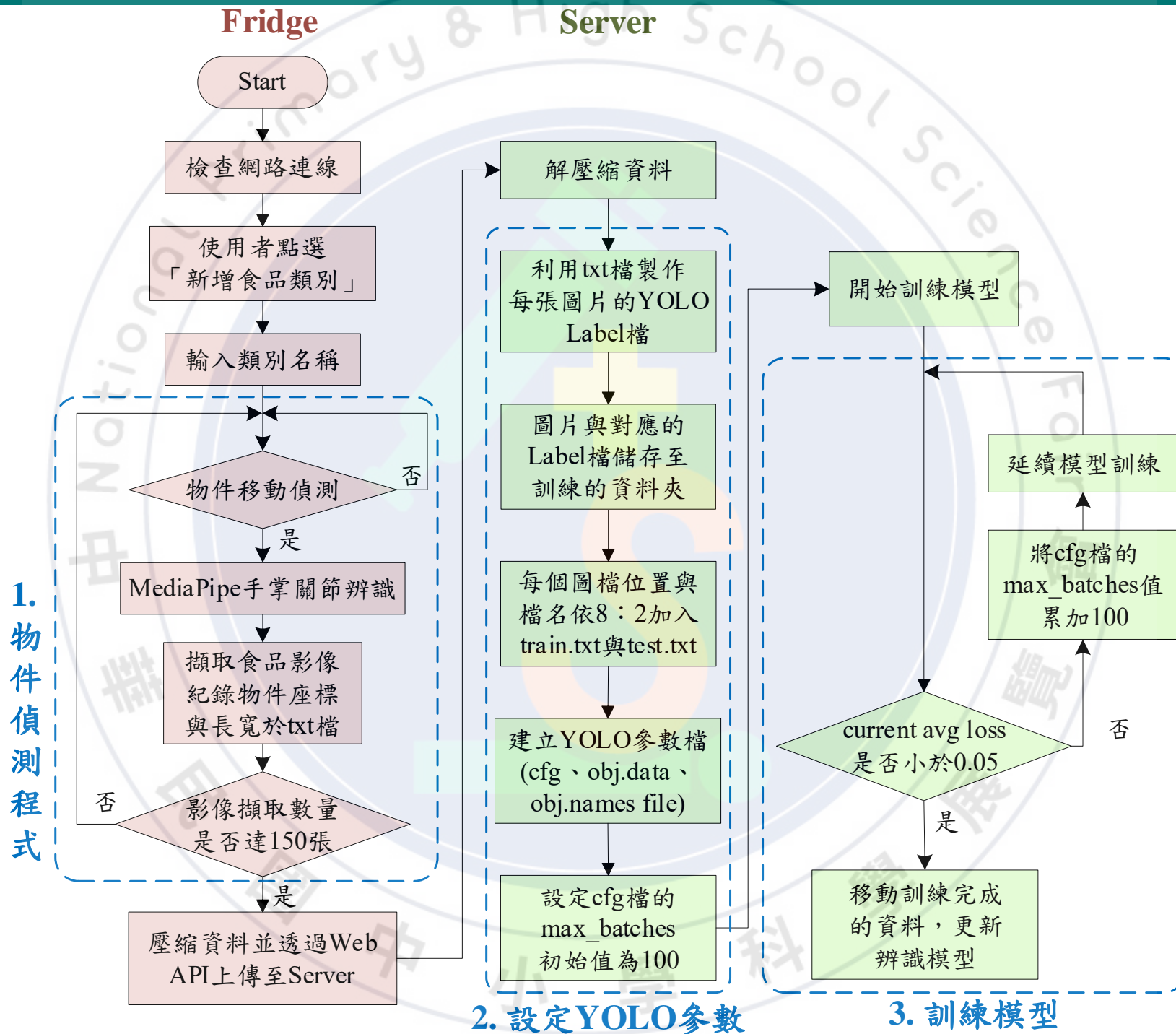


圖5：新增食品類別流程圖

研究方法-新增食品類別(2/3)

1. 物件偵測程式：

以Python撰寫的程式，比較每一影格與平均影像的差異，篩選變動程度大於門檻值的區域，將變動面積大於設定值的區域整合，形成圖7、圖8中綠框的部分。

使用MediaPipe套件以及訓練後的手部辨識模型，辨識畫面中手掌與各關節的位置，如圖7、圖8黑框與描繪出的關節點。離左右邊界最遠的掌指關節為參考點，如下圖7、圖8的藍色節點。



圖7：移動偵測與手掌排除 1

圖8：移動偵測與手掌排除2

1. 物件偵測程式

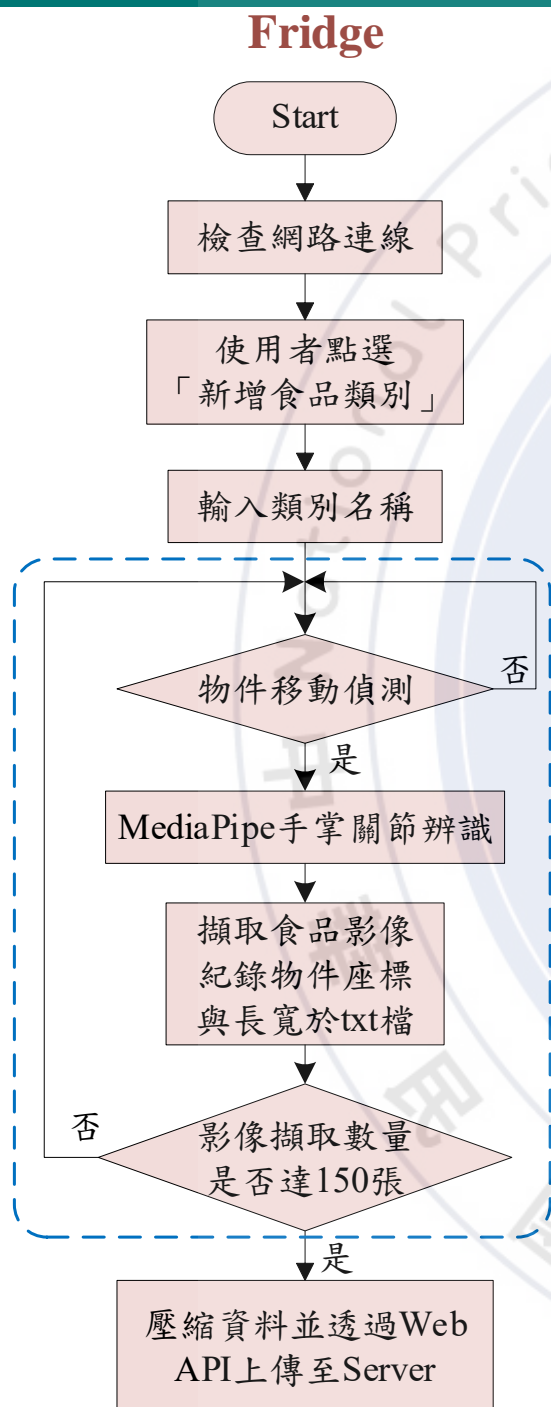
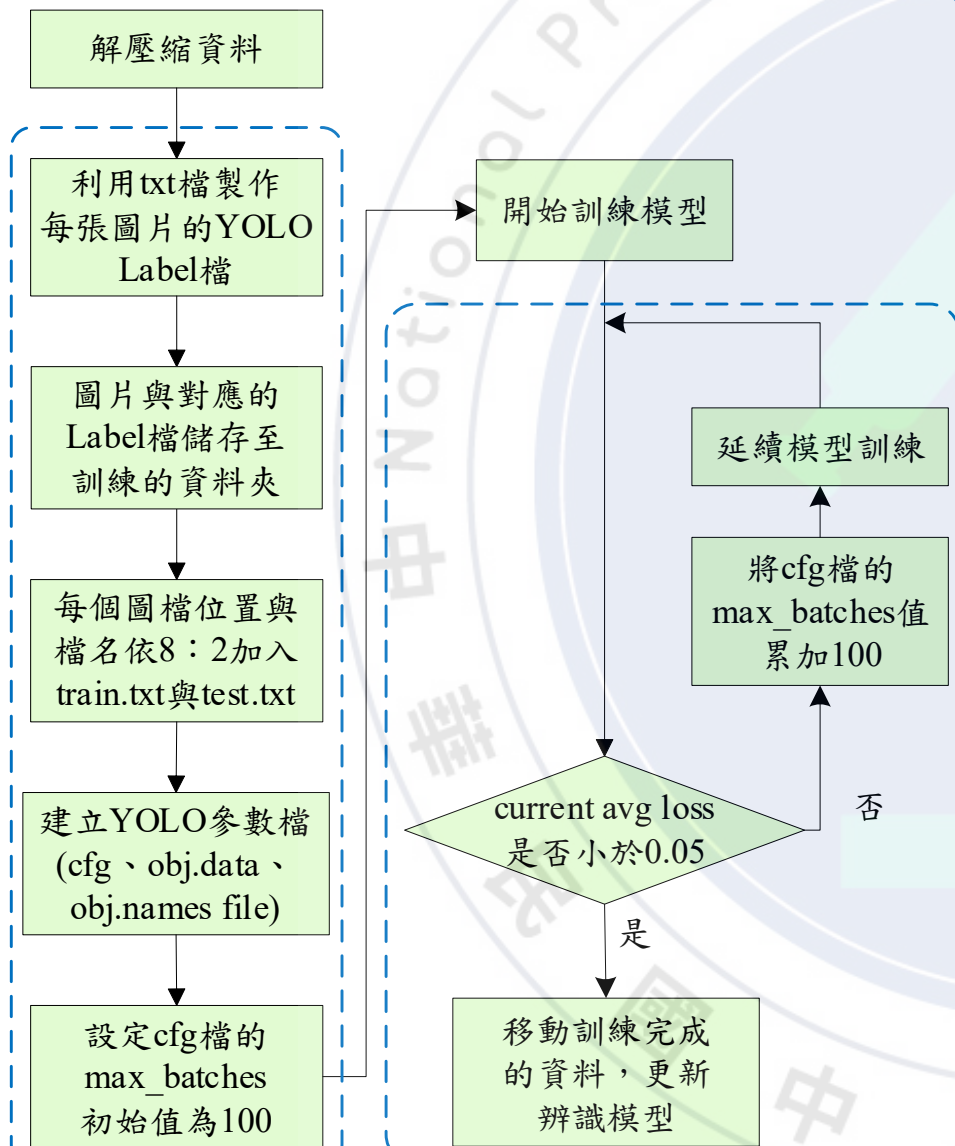


圖6：新增食品類別流程圖-左側

研究方法-新增食品類別(3/3)

Server



2. 設定YOLO參數

3. 訓練模型：訓練模型至current avg loss值低於0.05。圖10為weights的訓練疊代次數收斂圖，其中經歷9457次疊代後，可得current avg loss值0.2875，並且mAP(Mean Average Precision)達到97.4%。

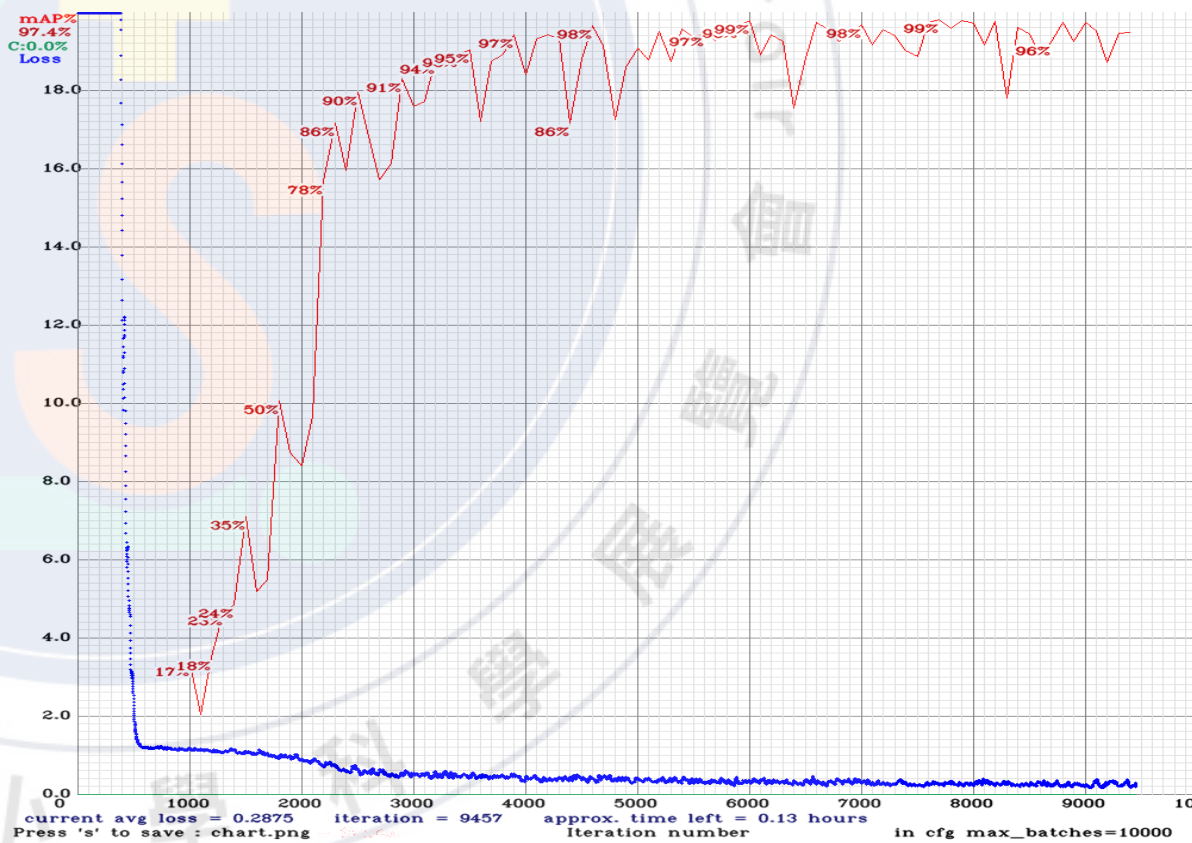


圖10：模型訓練mAP與Average Loss Rate

2. 設定YOLO參數

圖9：新增食品類別流程圖-右側

3. 訓練模型

研究方法-影像辨識放入與拿取食材

當冰箱偵測到冰箱門開啟時，冰箱會擷取開門期間的影像，利用Web API上傳至Server進行辨識食品品項與食品的移動軌跡，並且分析座標的x軸變化判斷放入或拿取，接續檢查食品清單或buffer食品清單中有沒有多個相同品項之食品。

若有多個相同食品，則詢問使用者再更新資料庫，無則直接更新資料庫。圖12、圖13為辨識食品類別準確度與座標位置。

tea: 71% (left_x: 373 top_y: 424 width: 346 height: 900)

Fridge

Server

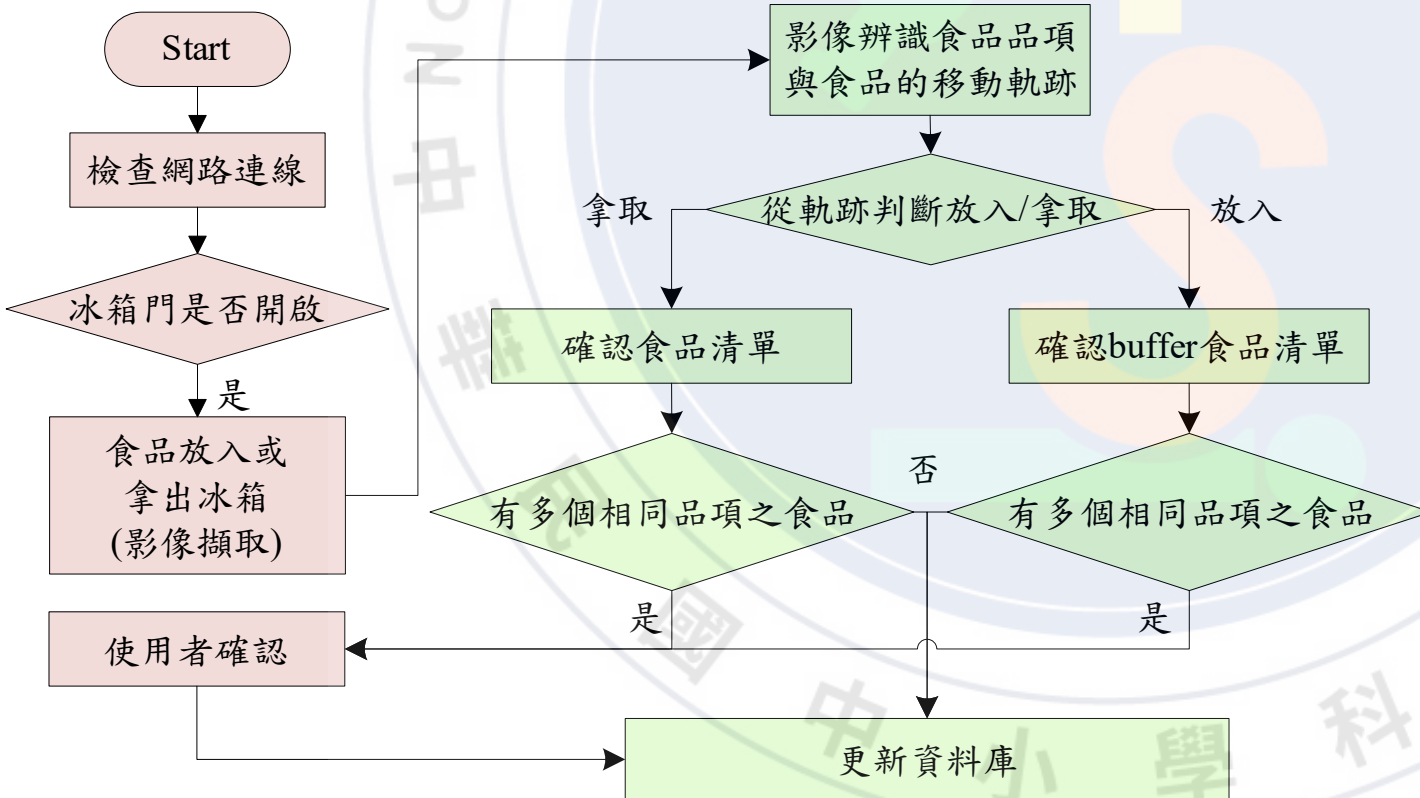


圖11：放入與拿取食品辨識

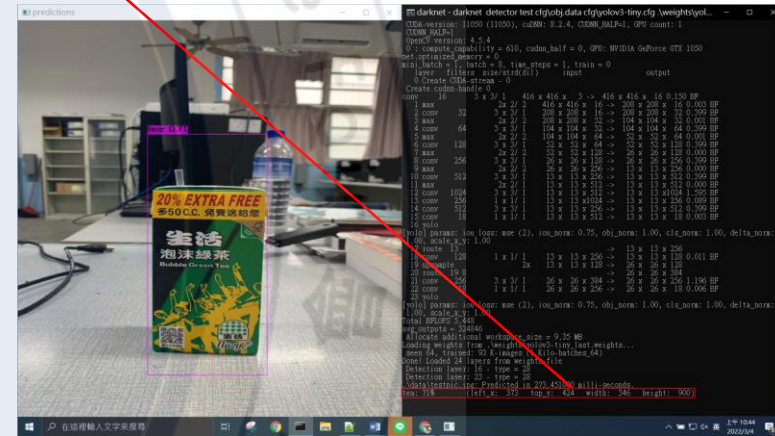


圖12：辨識食品類別準確度與座標位置

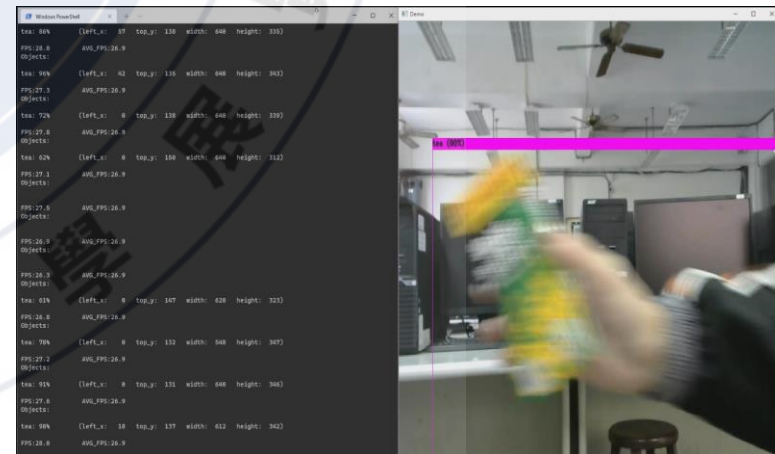


圖13：辨識食品移動軌跡

研究方法-既有類別增量訓練

1. 收集增量訓練資料

將每次拿取與放入所辨識的影片，篩選每影格可信度高於50%的辨識結果，將影片依品項與座標分別儲存與製作YOLO Label檔。

2. 設定YOLO參數

3. 訓練模型：繼續訓練原本的weights模型，至current avg loss值低於0.05。

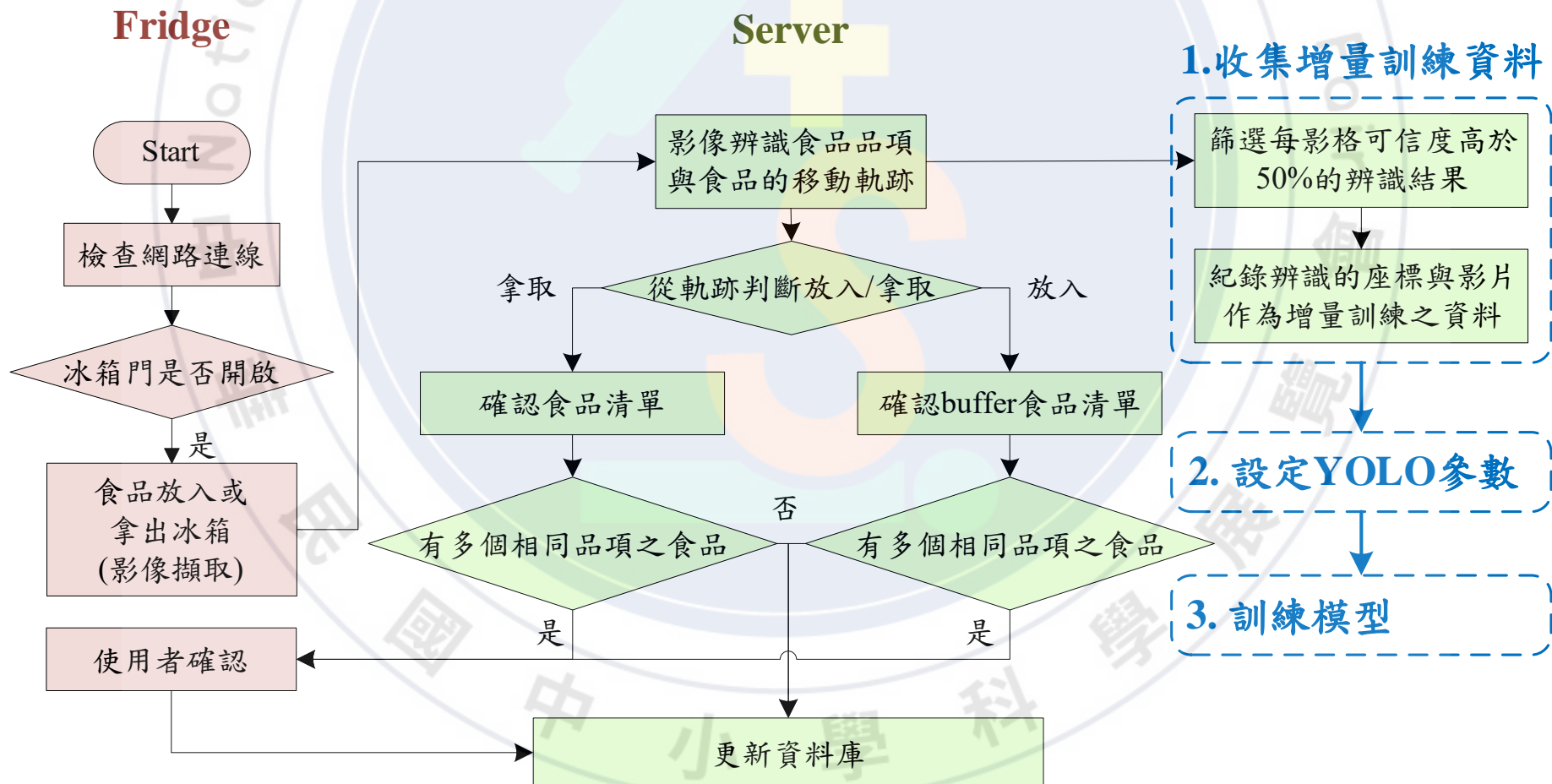


圖14：放入與拿取食品辨識、既有類別增量訓練流程圖

研究結果(1/2)

一、食材放入冰箱

使用者開啟冰箱門放入綠茶之流程，經由冰箱門內Camera 2擷取影片上傳至Server辨識，並依據Server之辨識結果更新食品清單。



圖15：開啟冰箱門



圖16：食品放入冰箱

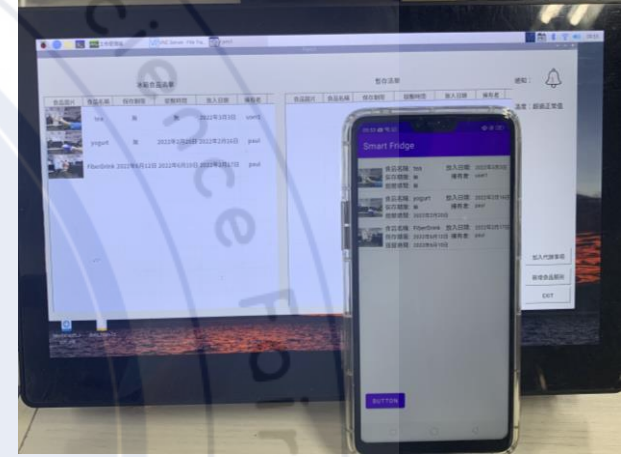


圖17：食品清單更新

二、食材拿出冰箱

當冰箱內有相同品項的多個食品，使用者拿取其中一樣時，系統會跳出辨識結果，並詢問使用者確切拿取的項目，使用者確認後，此項目會被放置到食品暫存清單。



圖18：拿取其中一個優格

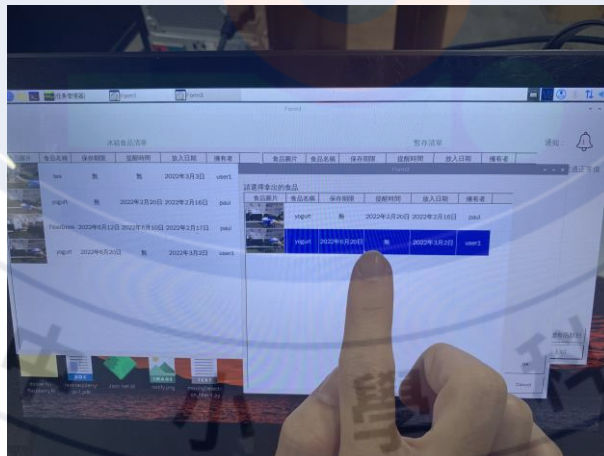


圖19：確認拿取的品項

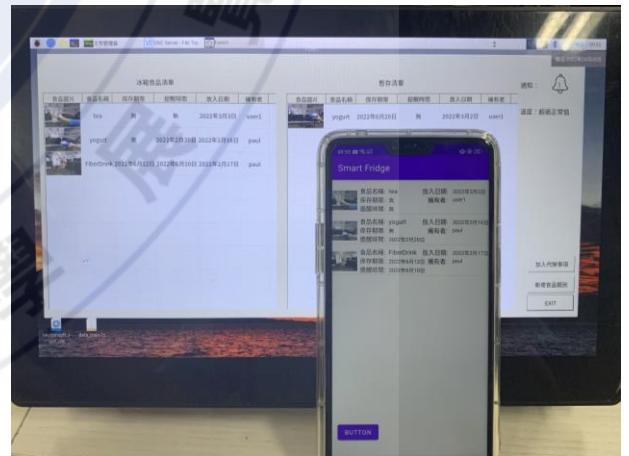


圖20：食品清單與暫存清單畫面

研究結果(2/2)

三、新增食品類別

使用者點選冰箱介面中「新增食品類別」的按鈕，並且輸入食品名稱，系統會跳出「物件偵測程式」擷取150張食品影像，並上傳至Server進行模型訓練。



圖21：點選新增食品類別按鈕

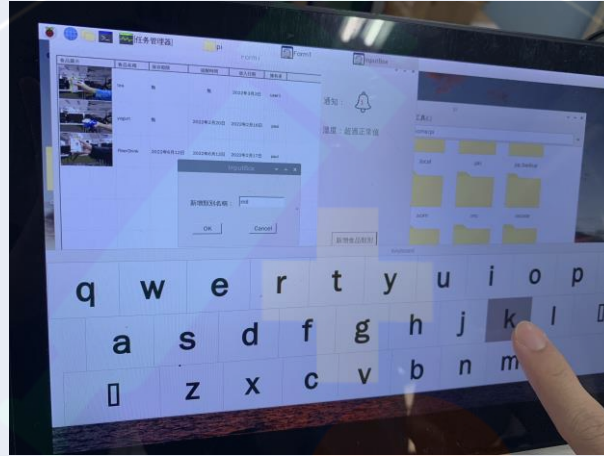


圖22：輸入食品類別名稱



圖23：冰箱擷取預備訓練之圖檔

討論

功能 \ 廠牌	小米	LG	三星	本系統
影像辨識食品	×	×	× (註 1)	✓
新增辨識食品類別	×	×	×	✓
紀錄食品保存期限	×	×	✓	✓
遠端查看食品	✓	✓	✓	✓
紀錄食品擁有者	×	×	×	✓
提醒食品過期	×	×	✓	✓

註1:需手動建檔

結論

本研究中提出了一個基於影像辨識且具學習能力的智慧冰箱系統：

- 可以新增食品類別，訓練影像辨識模型
- 自動化地建立與刪除冰箱食品清單
- 有效避免食材擺放的死角，導致無法辨識的問題
- 擷取拿取/放入食材之影像，作為增量訓練的資料，提高系統辨識準確度
- 可藉由APP與冰箱觸控螢幕查看與修改食品清單與狀態
- 可藉由APP與冰箱觸控螢幕顯示過期提醒與冰箱異常通知

未來可望進一步辨識食品的保存期限、增加對食品的營養分級、分析使用者購買習慣與賣場特價資訊，推薦菜單與自動訂購或外送平台之服務，為消費者節省時間與金錢，增進使用者體驗。

本系統不僅可用於智慧冰箱之情境，也可用於其他需要物件辨識與建檔記錄等領域，如：工廠庫房、物流倉儲...等，皆可使用本研究作為系統基礎架構來延伸使用。

參考資料

[1]胡豐榮、徐先正(譯)(2020年1月31日)。機器學習的數學基礎：AI、深度學習打底必讀(原作者：西內啟)。旗標(原著出版年：2020年1月31日)

[2] LG InstaView™ Door-in-Door® Refrigerators。LG。

[3] Keep track of what's in your smart fridge with family Hub。Samsung。