

# 朝陽科技大學

## 資訊與通訊系

### 【實務專題成果報告】

#### 蛋雞飼料調配及狀況監控系統

指導教授：鄭煜輝 博士  
組 員：徐千善 (10830074)  
廖恩彬 (10830018)  
陳昶渝 (10830056)  
黃宗櫟 (10830036)  
謝宗霖 (10830086)

2 0 2 2 年 1 2 月

# 朝陽科技大學專題報告授權同意書

本同意書所授權之報告為本組在朝陽科技大學資訊與通訊系111學年度第一學期修習之☐畢業專題/☒實務專題/☐專題設計/☐專題報告/☐其他。

專題名稱：蛋雞飼料調配及狀況監控系統

本組就具有著作財產權之報告全文資料，基於資源共享、合作互惠、回饋社會及促進學術研究之理念，同意無償、非專屬授權朝陽科技大學將本組報告不限地域、時間及次數，以紙本、光碟、網路或其它方式收錄、編輯、重製或發行，以供本校典藏，並提供讀者個人非營利性質之檢索、瀏覽、下載或列印。

本組擔保本著作係著作人之原創性著作，並未侵害任何第三人之智慧財產權。倘若若有抄襲行為進而造成侵害他人著作權之情形，願自行負擔相關法律責任。

請勾選電子全文檔授權公開年限：

☐ 3年後公開☐ 5年後公開

指導老師簽名：

學生簽名： \_\_\_\_\_ 學號： \_\_\_\_\_

(務必填寫)

中 華 民 國 年 月 日

## 目錄

|                              |    |
|------------------------------|----|
| 朝陽科技大學專題報告授權同意書 .....        | I  |
| 目錄 .....                     | II |
| 圖目錄 .....                    | IV |
| <b>1. 前言</b> .....           | 1  |
| <b>2. 緒論</b> .....           | 2  |
| 2.1 研究背景 .....               | 2  |
| 2.2 研究目的 .....               | 2  |
| 2.3 研究問題 .....               | 2  |
| 2.4 研究方法 .....               | 2  |
| 2.5 研究貢獻 .....               | 3  |
| <b>3. 研究方法</b> .....         | 3  |
| <b>4. 環境與系統架構</b> .....      | 5  |
| 4.1 環境 .....                 | 5  |
| 4.1.1 Windows 10 .....       | 5  |
| 4.2 軟體 .....                 | 5  |
| 4.2.1 PyCharm .....          | 5  |
| 4.2.2 Arduino IDE .....      | 6  |
| 4.2.3 Bootstrap5 .....       | 6  |
| 4.2.5 SQLite .....           | 7  |
| 4.2.6 YOLOv4 .....           | 7  |
| 4.3 硬體 .....                 | 7  |
| 4.3.1 Arduino UNO .....      | 7  |
| 4.3.2 28BYJ-48 步進馬達 .....    | 8  |
| 4.3.3 L298n 驅動馬達模組 .....     | 8  |
| 4.3.4 直流馬達 .....             | 8  |
| 4.3.5 HC-SR04 .....          | 9  |
| 4.3.6 Webcam .....           | 9  |
| 4.5 系統架構 .....               | 10 |
| <b>5. 系統功能介紹</b> .....       | 10 |
| 5.1 功能說明 .....               | 10 |
| 5.1.1 死雞辨識 .....             | 10 |
| 5.1.2 Line Notify 即時通知 ..... | 10 |
| 5.1.3 雞籠的影像串流及死亡紀錄 .....     | 10 |
| 5.1.4 自動飼料車 .....            | 10 |
| 5.2 操作流程 .....               | 11 |
| 5.2.1 網頁介面 .....             | 11 |
| 5.2.2 Line Notify .....      | 11 |

|                       |    |
|-----------------------|----|
| 5.2.3 自動飼料車控制介面 ..... | 12 |
| 5.2.4 自動飼料車 .....     | 12 |
| 6. 研究結果及討論 .....      | 13 |
| 7. 結論 .....           | 14 |
| 參考資料 .....            | 14 |

## 圖目錄

|                                  |    |
|----------------------------------|----|
| 圖 4.1.1.1 1 Windows 示意圖 .....    | 5  |
| 圖 4.2.1.1PyCharm 開發環境 .....      | 5  |
| 圖 4.2.2.1 Arduino IDE 開發環境 ..... | 6  |
| 圖 4.2.3.1 Bootstrap5 .....       | 6  |
| 圖 4.2.4.1 Django .....           | 6  |
| 圖 4.2.5.1 SQLite .....           | 7  |
| 圖 4.2.6.1 YOLOv4 .....           | 7  |
| 圖 4.3.1.1 Arduino UNO 開發版 .....  | 7  |
| 圖 4.3.2.1 28BYJ-48 步進馬達 .....    | 8  |
| 圖 4.3.3.1 L298n 驅動馬達模組 .....     | 8  |
| 圖 4.3.4.1 直流馬達 .....             | 8  |
| 圖 4.2.5.1 HC-SR04 .....          | 9  |
| 圖 4.3.6.1 Webcam .....           | 9  |
| 圖 4.5.1 系統架構圖 .....              | 10 |
| 圖 5.2.1.1 網頁示意圖 .....            | 11 |
| 圖 5.2.2.1 Line Notify 簡訊介面 ..... | 11 |
| 圖 5.2.3.1 飼料機控制 UI 介面 .....      | 12 |
| 圖 5.2.4.1 自動飼料車 .....            | 12 |

# 蛋雞飼料調配及狀況監控系統

鄭煜輝 徐千善 廖恩彬 陳昶渝 黃宗樑 謝宗霖

朝陽科技大學資訊與通訊系

yhcheng@cyut.edu.tw

## 摘要

近年來，隨著科技快速的進步，在自動養殖的應用越來越受到矚目，但在蛋雞養殖這一塊，仍然是非常稀少，仍需要依靠大量人力去執行養殖的動作，加上政府的政策導致沒有很多人去執行蛋雞自動化養殖場，因此本專題實現了遠端的雞隻監控並監測死雞傳送通知，以及飼料的投放器。透過遠端的操作方式，以此來減輕養殖人員的工作上的需求。

關鍵字：雞隻監控、死雞偵測、飼料車、影像辨識

## 1. 前言

目前國內的養殖場全都是以飼料車與人力傳統方式運作，再加上設備機器十分老舊，另外，完全依靠養殖人力採行人眼進行每個雞籠裡面辨識三至四隻雞隻存活的狀況，本次專題製作將以自動化飼料機及透過影像辨識養殖場雞隻中是否有死雞的情形。透過這樣方式進行可減少人力負擔、提升人工效能，徹底解決傳統飼料車餵養雞隻未盡周延的自動化監控所帶來的缺失；同時，透過遠距操作與影像辨識監控模式，除可解決長期使用大量人力檢查雞隻的問題，並經由系統記錄與通知達成雞隻餵養與清理的效能。

## 2. 緒論

### 2.1 研究背景

近年來，隨著科技的進步和自動化的應用，蛋雞場自動飼料車和辨識雞隻存活與死亡，已成為一個值得研究探討的議題。目前傳統的養殖場雞隻圈養通常需要大量的人力檢查每一隻雞的狀況，而其過程需要將雞拉出來進行檢查，其流程既繁瑣的又和高度需要人力，使得傳統的方法實不再符合現代化的需求。自動飼料車，使用資訊化方式操控電機或其他動力移動車身，僅需人力設定操作設備即可完成之前全人力餵養與篩檢工作，大大提升了效率。同時，因自動飼料車也可以透過遠端操控進行自動化控制系統來實現飼料的投放，使得餵養者在操作使用更加方便。因此，本研究的目的是設計蛋雞場自動飼料車及利用影像辨識技術來解決傳統的死雞辨識方法的不足，配合透過自動化的控制和遠端操控的方式，提升飼料車的效率和便利性，以及提升死雞辨識的準確度和效率，期為未來的自動化養殖提供參考。

### 2.2 研究目的

透過影像辨識技術，提升死雞辨識的準確度和效率，以及設計蛋雞場自動飼料車，透過自動化的控制和遠端操控的方式，提升飼料車的效率和便利性，以減輕人力負擔，提升養殖場的效率，並為未來的自動化養殖提供參考。此外，透過設計遠端操控系統，本研究也希望能夠減輕養殖人員的在工作上的負擔，並提升工作效率。

### 2.3 研究問題

在死雞辨識方面：如何使用影像辨識技術來提升死雞辨識的準確度？如何設計影像辨識系統，使其能夠有效地辨識出各種不同的死雞狀況？在多種不同的環境下，影像辨識系統的準確度是否有所提升？如何解決影像辨識系統在某些情況下的辨識問題，例如光照不足、雞隻擠在一起等情況？

在自動飼料車方面：如何設計蛋雞場自動飼料車，使其能夠有效地提升飼料車的效率和便利性？如何設計遠端操控系統，使養殖人員能夠輕鬆地進行飼料車的操作？如何設計飼料車的車身，使其能夠有效地進行飼料的投放？

### 2.4 研究方法

在死雞辨識方面，本研究將使用影像辨識技術來進行研究。首先，將收集大量的雞的影像數據，包括死雞和活雞，並將數據分為訓練數據和測試數據。接下來，將利用深度學習技術對訓練數據進行訓練，建立影像辨識模型。在訓練過程中，將適當地調整模型的參數，以使模型能夠更好地擬合數據。最後，將利用測試數據對辨識模型進行驗證，評估其準確度。

在自動飼料車方面，本研究將採用實驗設計法進行研究。首先，將設計出多種不同的自動飼料車模型，並對其進行模擬實驗。在模擬實驗中，將測量不同模型的飼料

車的效率和便利性。接下來，將選擇其中表現較佳的模型進行實際測試。在實際測試中，將記錄飼料車的運行狀況，並對其進行評估。

## 2.5 研究貢獻

在死雞辨識方面，提供了一種新的方法來解決傳統的死雞辨識方法的不足，使得死雞辨識的準確度和效率得到大幅提升。透過對影像辨識模型的訓練，可以使得辨識系統能夠更好地適應各種不同的環境，例如光照不足、雞隻擠在一起等情況。在自動飼料車方面，提供了一種新的自動飼料車設計，可以提升飼料車的效率和便利性，減輕人力負擔。設計了遠端操控系統，使得養殖人員能夠輕鬆地操作飼料車，提升工作效率。本研究的結果可以為未來的自動化養殖提供參考，幫助養殖場更好地管理雞隻的健康狀況，以及為未來的自動化養殖提供參考，幫助養殖場更好地管理雞隻的飼料和餵食。

## 3. 研究方法

在本研究中，我們使用了 YOLOv4 演算法來進行影像辨識。YOLO (You Only Look Once) 是一種快速的影像辨識演算法，可以對影像進行快速的目標檢測。首先，我們使用 Python 程式語言並安裝必要的套件，包括 OpenCV、NumPy 和 YOLOv4 等。接下來，我們收集了 2500 張的雞的影像，一共有 5 個場景，再透過 LabelImg 影像標註工具製作深度學習資料集，並將其分成訓練集和測試集。訓練集用於訓練影像辨識模型，測試集用於評估模型的準確度。

接下來，我們使用 YOLOv4 的訓練工具來訓練影像辨識模型。我們首先下載 YOLOv4 的預先訓練模型，並將其用於我們的訓練集。我們選擇了較小的訓練模型，以減少訓練時間。我們設定了訓練的參數，包括學習率、批次大小和訓練輪數等。在訓練過程中，我們監測訓練損失和準確度，並在訓練結束後保存最佳模型。在訓練結束後，我們將使用測試數據來評估所建立的模型的準確度。我們將收集大量的死雞和活雞的影像數據，並將其作為測試數據。然後，我們將利用 YOLOv4 架構對測試數據進行預測，並對預測結果進行評估。我們將使用不同的評估指標，例如精確度、召回率和 F1 分數，來評估模型的表現。此外，我們還將比較不同的訓練方法對模型表現的影響，並探討在不同的環境下模型的表現情況。這些信息將有助於我們更好地了解 YOLOv4 在死雞辨識方面的潛力，並指導我們在未來的研究中進一步提升模型的表現。

我們使用網站框架 Django 與 YOLOv4 深度學習演算法結合實現影像辨識功能。首先，我們將使用 Django 開發網站前端界面，使用者可以在網站上傳影像並且觀看辨識結果。接著，我們將使用 YOLOv4 深度學習演算法對傳入的影像進行辨識。



在訓練結束後，我們將使用影像串流的技術將辨識結果即時傳回給使用者。最後，我們將對系統進行測試，評估其準確度及效率。

我們利用 Django 框架開發網站前端，使用者可以透過網站傳送影像並查看辨識結果。我們將使用 YOLOv4 深度學習演算法對影像進行辨識，並使用影像串流的技術將辨識結果即時傳送回網站。首先，我們會先透過 Django 的開發環境建立網站架構並設計網頁版面。接著，我們將使用 YOLOv4 的訓練模型對影像進行辨識，並使用 OpenCV 對影像進行處理。最後，我們使用影像串流的技術將辨識結果即時傳送回網站，使用者可以透過網站查看辨識結果。透過這個研究方法，我們希望能夠提供一個方便使用者查看辨識結果的網站，並提升辨識的準確度和效率。在死雞辨識系統方面，我們使用 Line Notify 即時收到死雞訊息，以即時解決死雞問題。

在自動飼料車方面，我們使用自製的自動飼料車模型來進行實驗。我們首先設計了多種不同的自動飼料車模型，並對其進行模擬實驗，選擇其中表現較佳的模型進行實際測試。然後使用 Arduino 控制板作為自動飼料車的核心控制裝置。我們使用直流馬達來控制移動軸，使自動飼料車能夠自主移動。測量飼料量的方法使用超音波測距技術來測量飼料道中的飼料量。首先，我們將在飼料車上安裝超音波測距感測器，並對其進行校正。接下來，在飼料車進行飼料投放時，感測器將會不斷測量飼料道中的飼料量。我們將利用超音波測距感測器測量的數據，來控制飼料車的飼料投放量，使得飼料道中的飼料量保持在理想的水平。透過使用超音波測距技術來測量飼料道中的飼料量，我們可以更精確地控制飼料車的飼料投放量，並提高飼料車的效率和便利性。

此外，我們將使用 Tkinter GUI 庫來製作使用者介面 (UI)。Tkinter GUI 庫提供了許多用於創建 UI 的工具和控件，使我們能夠輕鬆地創 UI 介面，並且易於使用。同時我們使用 Pyserial 進行電腦與 Arduino 的連接，以便在收到控制指令後，能夠有效地控制飼料車的動作。

## 4. 環境與系統架構

### 4.1 環境

#### 4.1.1 Windows 10

Windows 10 提供了多種功能，可以滿足不同用戶的需求。例如，Windows 10 支持多任務運行，可以同時運行多個應用程序；它還提供了各種檔案管理工具，可以方便用戶管理電腦中的檔案和資料。



圖 4.1.1.1 Windows 示意圖

### 4.2 軟體

#### 4.2.1 PyCharm

PyCharm 是一個功能強大的 Python 開發環境 (IDE)，由 JetBrains 公司開發。它提供了豐富的工具和功能，可以讓 Python 開發人員更加高效、輕鬆地進行程序開發。

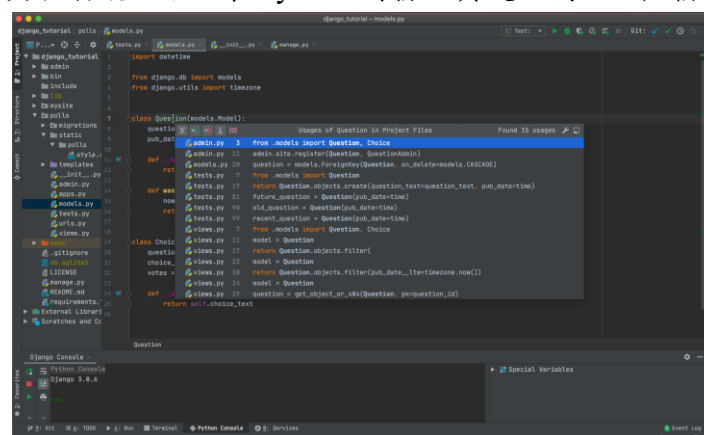


圖 4.2.1.1PyCharm 開發環境

## 4.2.2 Arduino IDE

Arduino IDE 是 Arduino 微控制器的官方開發環境（IDE）。它是一款免費、開源的軟件，可以讓開發人員輕鬆地編寫、測試和上傳程序到 Arduino 微控制器。

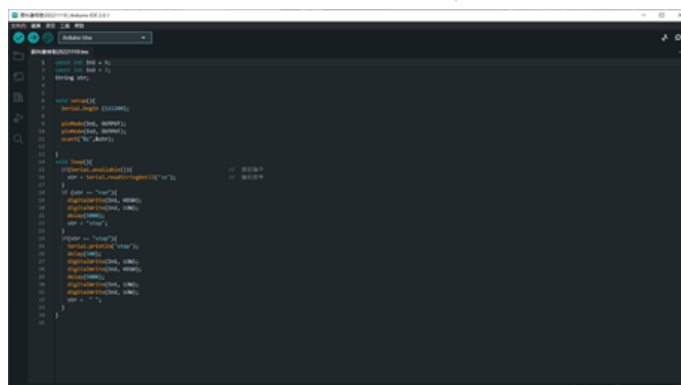


圖 4.2.2.1 Arduino IDE 開發環境

## 4.2.3 Bootstrap5

Bootstrap 是一個前端框架，可以讓開發人員快速地構建網站和應用。它提供了豐富的 HTML、CSS 和 JavaScript 組件，可以讓開發人員輕鬆地構建響應式、移動優先的網站。



圖 4.2.3.1 Bootstrap5

## 4.2.4 Django

Django 是一個 Web 應用框架，可以用於構建各種 Web 應用。它使用 Python 語言編寫，提供了許多實用的功能，例如模板系統、表單系統、管理後臺等，使開發人員能夠快速、高效地構建 Web 應用。

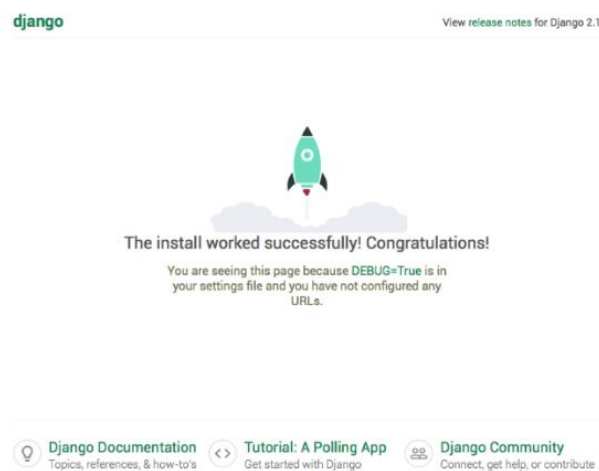


圖 4.2.4.1 Django

#### 4.2.5 SQLite

SQLite 是一個輕量級的關係資料庫管理系統，非常適合用於嵌入式設備和移動應用。它具有輕巧、快速、低功耗等優點，使得 SQLite 成為許多應用的首選資料庫。



圖 4.2.5.1 SQLite

#### 4.2.6 YOLOv4

YOLOv4 是一種計算機視覺模型，用於對圖像中的目標進行分類、定位和識別。它是 YOLO（You Only Look Once）模型。YOLOv4 在性能上有顯著的提升，具有更快的速度和更高的精度。同時，它還支援多種新的架構和優化方法，使得它能夠在更複雜的情況下仍然取得良好的性能。



圖 4.2.6.1 YOLOv4

### 4.3 硬體

#### 4.3.1 Arduino UNO

我們使用 Arduino UNO 控制板是因為我們要用來控制飼料機動作，相較於 Arduino Mega 控制板來說，Mega 控制板的效能溢出、與用的端口也不用那麼多，而且價格也較貴。



圖 4.3.1.1 Arduino UNO 開發版

#### 4.3.2 28BYJ-48 步進馬達

我們這邊使用 28BYJ-48 步進馬達是用來控制我們飼料量投放的大小。

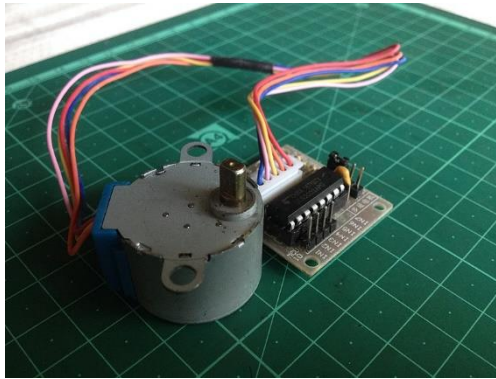


圖 4.3.2.1 28BYJ-48 步進馬達

#### 4.3.3 L298n 驅動馬達模組

我們使用這一個直流馬達驅動板來控制我們直流馬達前進或後退。

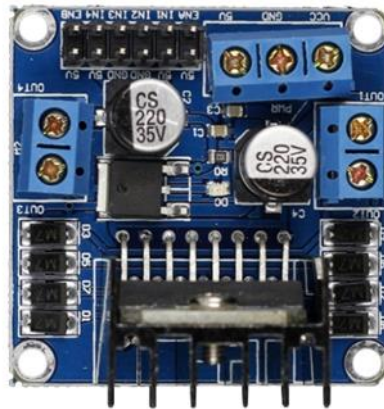


圖 4.3.3.1 L298n 驅動馬達模組

#### 4.3.4 直流馬達

我們使用本顆直流馬達來控制飼料機的移動功能。由於移動的物品有一定的重量，所以選擇這一顆 9-12V 的直流馬達。



圖 4.3.4.1 直流馬達

#### 4.3.5 HC-SR04

HC-SR04 是一種超聲波測距感測器，可以用於測量距離。它通過發射超聲波訊號並測量返回訊號的時間差來求出距離。



圖 4.2.5.1 HC-SR04

#### 4.3.6 Webcam

鏡頭的部分是隨便鏡頭都可以，為了節省金費所以使用本來就擁有的羅濟 C922 PRO HD STREAM WEBCAM 鏡頭來做實驗。



圖 4.3.6.1 Webcam

#### 4.4 通訊協定/設備:UART、HTTPS

## 4.5 系統架構

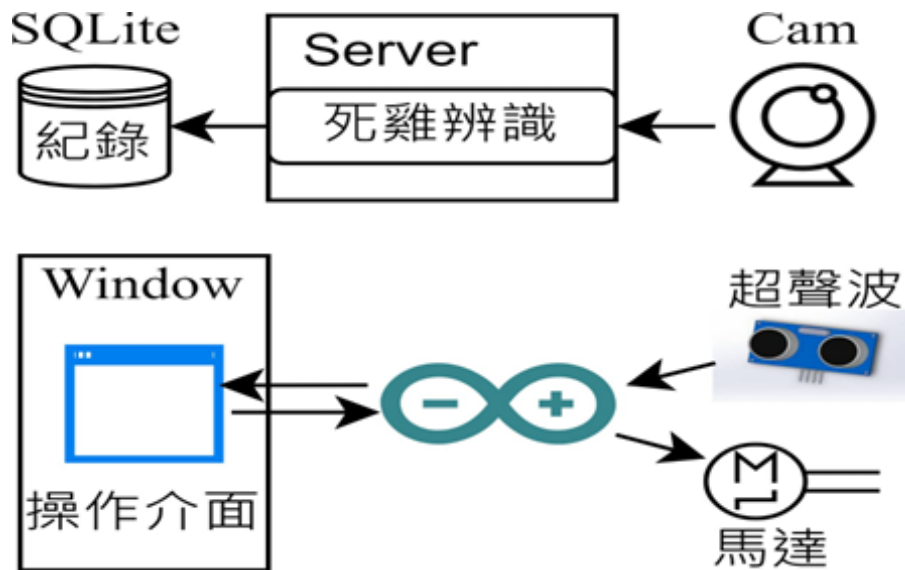


圖 4.5.1 系統架構圖

## 5. 系統功能介紹

### 5.1 功能說明

#### 5.1.1 死雞辨識

每一段時間進行一次死雞辨識。判斷雞籠裡面是否存在死雞。以達到減輕養殖人員工作量的目的。

#### 5.1.2 Line Notify 即時通知

每當偵測到死雞時，會馬上傳送訊息至 Line Notify，讓養殖人員可以在第一時間進行清理。

#### 5.1.3 雞籠的影像串流及死亡紀錄

我們在網站中設置了雞籠的一個影像串流，同時在下方設置了一個死亡紀錄的一個區塊，方便飼養員觀察雞籠的狀況，以及瀏覽紀錄。

#### 5.1.4 自動飼料車

我們為飼料車設計了兩種模模式，一個是一般模式，一般模式顧名思義就是在一班正常的時候使用的一個模式。另一個是退羽模式，這個模式是在雞隻退羽時期使用的一個模式。

## 5.2 操作流程

### 5.2.1 網頁介面

啟動我們啟動服務器後，可以在看到在我們在網站上方有一個即時的影像串流畫面，以及下方的過往紀錄，並且同時會啟動死雞偵測系統，如圖 5.2.1.1。

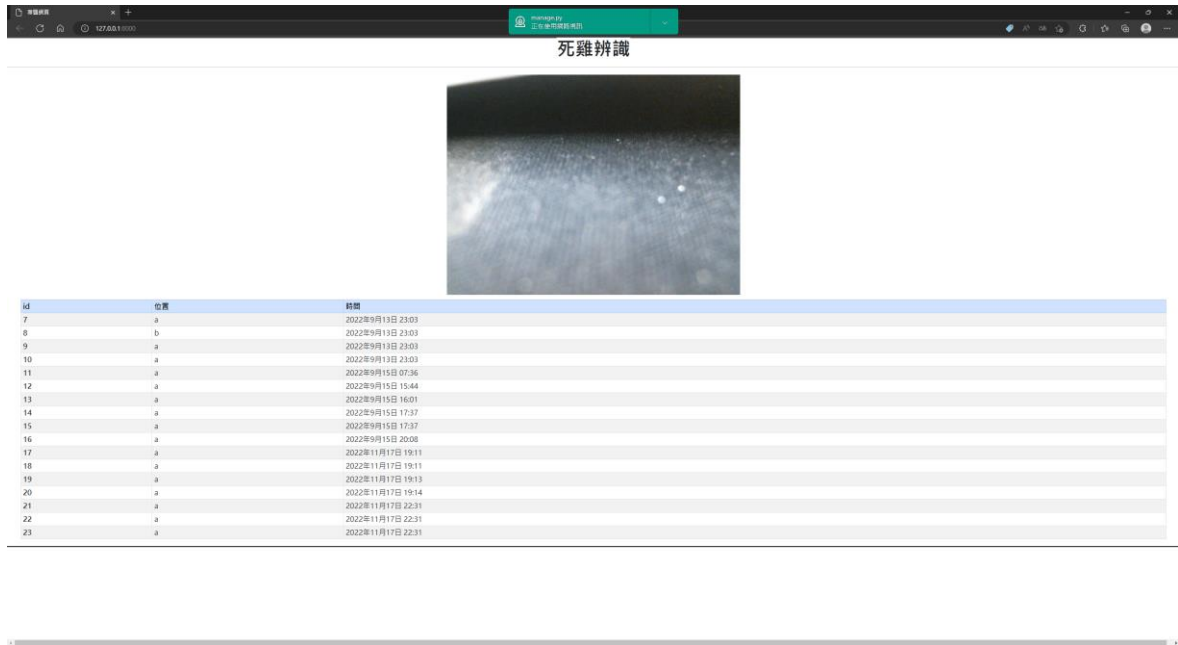


圖 5.2.1.1 網頁示意圖

### 5.2.2 Line Notify

在我們系統辨識偵測到死雞時，會傳送一通簡訊到我們的 Line Notify 提示我們說雞籠裡面有死亡的雞隻需要趕快去清理，如圖 5.2.2.1。

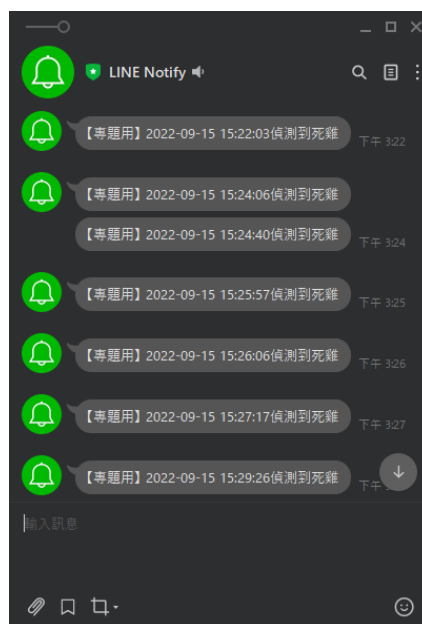


圖 5.2.2.1 Line Notify 簡訊介面



### 5.2.3 自動飼料車控制介面

我們開啟字弄飼料車程式後，選擇 UI 介面中的模式點及下去即可運行，如圖 5.2.3.1。



圖 5.2.3.1 飼料機控制 UI 介面

### 5.2.4 自動飼料車

當點擊按鈕後，如圖 5.2.4.1 飼料車會透過超音波進行飼料道中的飼料量偵測，然後控制飼料量馬達的速度。並進行來回的移動。



圖 5.2.4.1 自動飼料車

## 6. 研究結果及討論

在本研究中，我們使用 YOLOv4 演算法進行影像辨識，並收集了 2500 張雞的影像進行標註。我們將這些影像分成訓練集和測試集，並使用訓練集來訓練影像辨識模型。

經過訓練後，我們使用測試集對影像辨識模型進行評估。結果顯示，我們的影像辨識模型在測試集上的準確度為 98%。這表示，我們的模型能夠在大部分情況下正確地辨識雞的影像。但是，當影像曝光度過高或畫質過低時，辨識準確度可能會降低。這是因為在這些情況下，影像中的資訊會變得不足，使得模型無法正確地辨識。

因此，在實際應用中，我們需要注意影像的曝光度和畫質，以確保影像辨識的準確度。同時，我們還可以考慮使用更先進的影像辨識演算法，或是對訓練資料進行更多的增強，以提高模型的準確度。

此外，我們可以對模型進行參數調整或是使用不同的訓練方法，來提高模型的準確度。例如，我們可以使用更多的訓練資料來訓練模型，或是使用較大的神經網路架構來提高模型的辨識能力。

另外，我們還可以對訓練資料進行更多的增強，例如對訓練資料進行旋轉、縮放等，以更好地模擬真實的情況。透過這些方法，我們可以在一定程度上提高模型的準確度。

同時，我們還可以對模型的驗證方法進行改進，例如增加測試資料的數量或是使用不同的測試方法，以便更精確地評估模型的性能。透過這些方法，我們可以在一定程度上提高模型的準確度，並更好地滿足實際應用的需求。

在自動飼料車方面，我們使用 Arduino 控制板作為自動飼料車的核心控制裝置，並使用直流馬達來控制移動軸。我們使用超音波測距技術來測量飼料道中的飼料量，並使用這些數據來控制飼料車的飼料投放量。實驗結果顯示，我們的自動飼料車能夠自主移動並準確地投放飼料，提高了飼料車的效率和便利性。

此外，我們還可以對飼料車的控制系統進行升級，使用深度學習演算法來控制飼料車的飼料投放量，從而提高飼料車的效率和準確度。透過這些改進措施，飼料車在未來將會更加智能化，並且能夠更好地滿足實際應用的需求。

## 7. 結論

綜上所述，我們透過本研究強化了雞隻的監控能力並提升了飼料車的便利性。

在雞隻監控方面，我們透過遠端的方式來監測雞隻的生命訊號，並能夠及時發現死雞並通知養殖人員。這樣一來，可以有效地減輕養殖人員的工作壓力，並提升雞隻的生存率。

在飼料車方面，我們使用超音波測距感測器來測量飼料道中的飼料量，並利用 Arduino 控制板控制飼料車的動作。透過這種方式，我們可以更精確地控制飼料車的飼料投放量，並提升飼料車的效率和便利性。此外，我們使用 Tkinter 庫製作了使用者介面，使得操作更為方便。

在未來，我們可以透過影像辨識技術來改進飼料車的功能，讓它能夠自動偵測飼料量，達到更加自主的飼料投放。

總的來說，本研究成功地強化了蛋雞自動化養殖場的監控能力和飼料車的便利性，並且在未來仍有潛力進一步改進和提升。

## 參考資料

- [1] Mikechenx. (2021-11-11). YOLO v4 模型訓練實作. Retrieved from <https://ithelp.ithome.com.tw/articles/10282549?sc=pt>.
- [2] Liu, H.-W., Chen, C.-H. (ORCID), Tsai, Y.-C. (ORCID), Hsieh, K.-W., & Lin, H.-T. (2021, May 21). Identifying Images of Dead Chickens with a Chicken Removal System Integrated with a Deep Learning Algorithm. Department of Bio-Industrial Mechatronics Engineering, National Chung Hsing University, Taichung 402, Taiwan. <https://www.mdpi.com/1424-8220/21/11/3579>
- [3] 胡肄农, 窦少春, 王立方, 陆昌华. (2002). 蛋鸡规模化养殖场的自动监控系统. 江苏农业学报, (3). 江苏省农业科学院农业资源与环境研究中心, 扬州市银河科技公司. Retrieved from <http://www.cqvip.com/qk/92499x/200203/6821028.html>
- [4] 李卫国, 颜斌, 储新生. 蛋鸡规模养殖现状调查报告[J]. 中国畜禽种业, 2022, 18(1): 12-13. <http://xq.aiijournal.com/CN/>
- [5] 王玲, 王超峰, 郑奎. 基于 ARM 的蛋鸡养殖场网络视频监控系统设计[J]. 农业机械学 报, 2012(2): 186-191. Retrieved from <http://www.cqvip.com/qk/90304x/201202/40969093.html>
- [6] 燒飯程序猿. (2019.9.16). [python] PyCharm 入門教學 建立第一個專案 Hello World. <https://shofan.blogspot.com/2019/09/python-pycharm.html>

- [7] seaotter. (2021). Day 23: Tkinter-利用 Python 建立 GUI(基本操作及佈局篇). Retrieved from <https://ithelp.ithome.com.tw/m/articles/10278264>
- [8] cubie (2018). 使用 Python 的 pySerial 模組進行序列通訊：連接電腦與 Arduino 和 MicroPython. [online] 取自: <https://swf.com.tw/?p=1188>.
- [9] 維基百科 (2021). 螺旋 (簡單機械). 網址: [https://zh.m.wikipedia.org/zh-tw/%E8%9E%BA%E6%97%8B\\_\(%E7%B0%A1%E5%96%AE%E6%A9%9F%E6%A2%B0\)](https://zh.m.wikipedia.org/zh-tw/%E8%9E%BA%E6%97%8B_(%E7%B0%A1%E5%96%AE%E6%A9%9F%E6%A2%B0))
- [10] 威利斯 (2020). Day 8 : LINE Notify 訊息推播通知 (Python 版). 網址: <https://ithelp.ithome.com.tw/articles/10234115>
- [11] W. Chiu, "手把手程式實作分享系列：Django SQLite 資料庫設定集簡單操作," Bandai 的機器學習筆記, Aug 18, 2019. [Online]. Available: <https://medium.com/bandai%E7%9A%84%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E7%AD%86%E8%A8%98/%E6%89%8B%E6%8A%8A%E6%89%8B%E7%A8%8B%E5%BC%8F%E5%AF%A6%E4%BD%9C%E5%88%86%E4%BA%AB%E7%B3%BB%E5%88%97-django-sqlite%E8%B3%87%E6%96%99%E5%BA%AB%E8%A8%AD%E5%AE%9A%E9%9B%86%E7%B0%A1%E5%96%AE%E6%93%8D%E4%BD%9C-359529fdebe>
- [12] Yehchitsai (2021). AI & Data 觀賞魚辨識的 YOLO 全餐系列 第 22 篇 Day 22 - Django + YOLO 後台整合應用. Retrieved from <https://ithelp.ithome.com.tw/articles/10269994?sc=rss.iron>
- [13] TSAI, CEILING. (2013.4.6). Arduino 筆記(8)：控制步進馬達. [Blog post]. Retrieved from <https://atceiling.blogspot.com/2013/04/arduino.html>
- [14] GuanFuXinCSDN. (2020). Arduino 实践详细说明 (三) L298N 驱动直流电机. Retrieved from <https://blog.csdn.net/GuanFuXinCSDN/article/details/104158512>