

國立清華大學
工業工程與工程管理學系

智慧化企業整合
Intelligent Integration of Enterprise

饗饗 INPARADISE 吃到飽自助餐廳

基於深度學習之
空餐盤辨識與廚餘分析系統

第 5 組

108034069 黃靖珽

111034401 林明聰

111034561 郭彥佐

指導教授：邱銘傳 博士

中華民國 一 一 二 年 六 月

目錄

目錄	2
壹、簡介	3
一、 背景與動機	3
二、 研究目的	3
三、 5W1H 分析	4
四、 資料集介紹	4
貳、研究方法	5
一、 方法介紹	5
1. Yolo v5	5
2. Yolo v7	5
二、 資料前處理	7
三、 模型建立與訓練	8
1. Yolo v5	8
2. Yolo v7	14
叁、網頁頁面與功能介紹	18
一、網頁管理者系統及其頁面介紹	18
肆、結論	30

壹、簡介

一、背景與動機

在現代社會，隨著人們生活水平的提高，餐飲業的發展蓬勃，吃到飽餐廳也越來越受到人們的歡迎。然而，隨著客人數量的增加和菜品供應的豐富，吃到飽餐廳面臨著一個重要的問題，即有效收集客人使用完畢的空餐盤。傳統上，吃到飽餐廳依賴服務人員手動收集空餐盤，這樣不僅浪費了大量的人力資源，還可能導致餐廳服務效率的下降。因此，開發一種自動辨識空餐盤的系統將對提高吃到飽餐廳的運營效率和顧客滿意度具有重要意義。

此外，隨著社會對環境保護和可持續發展的關注增加，處理和管理廚餘成為一個迫切的議題。在吃到飽餐廳中，大量的廚餘產生，而廚師們需要了解廚餘的成分和數量，以便進行食物製備和口味調整。因此，開發一種能夠同時辨識空餐盤和分析廚餘成分的系統將為廚師們提供寶貴的數據和洞察，從而幫助他們更好地管理廚餘和減少食物浪費。

二、研究目的

基於以上背景，本研究旨在開發一個辨識吃到飽餐廳中客人所使用完畢的空餐盤的系統，並同時分析空餐盤中的廚餘成分。透過影像辨識技術和機器學習算法，我們的目標是實現自動化的餐盤收集，以提高餐廳的運營效率和顧客體驗。同時，我們希望提供準確的廚餘分析，幫助廚師們調整食物的製備和口味，從而實現食物資源的有效利用和減少廚餘的目標。

總之，本研究旨在結合影像辨識技術和廚餘分析，開發一個綜合解決方案，能夠同時辨識吃到飽餐廳中的空餐盤並分析其中的廚餘成分。這將有助於改善吃到飽餐廳的運營效率，減少人力成本，提供寶貴的數據支持給廚師們，同時也符合環境保護和可持續發展的要求。

三、5W1H 分析

When	備餐與用餐時段
Who	饗饗餐廳員工和顧客
What	辨識客人使用完畢的空餐盤，並且判斷其中廚餘的種類
Why	解決吃到飽餐廳面臨的廚餘問題，幫助廚師們調整食物供應和口味，提高資源使用效率，並減少環境浪費
Where	饗饗 INPARADISE 吃到飽自助餐廳
How	透過影像辨識和機器學習技術，開發系統來辨識空餐盤和判斷其中的廚餘種類。

四、資料集介紹

本研究使用「Roboflow」的線上資料集平台中的 Food Waste Detection 專案資料集，該資料集包含了 2716 張照片，並且在每張照片都附有物件分類的標記 label。下載後的 label 以座標的方式記錄在文字檔案中。圖片與其 label 如下圖所示：



圖 1. 資料集圖片範例

貳、研究方法

一、方法介紹

1. Yolo v5

YOLOv5 是一種基於深度學習的影像分類辨識方法，它在目標檢測和定位方面具有出色的性能和效率。該方法基於 YOLO 系列（You Only Look Once）的概念，採用了單階段的目標檢測策略，將目標的定位和分類合併為一個單一的預測過程。

YOLOv5 的核心思想是使用深度卷積神經網絡（CNN）進行特徵提取和分類。它使用了一個特殊的網絡結構，稱為特徵金字塔網絡（FPN），以獲得不同尺度的特徵圖。這使得 YOLOv5 能夠在不同大小的目標上進行有效的檢測和辨識。

此外，YOLOv5 還採用了一種稱為"Bag of Freebies"的訓練策略，以增強模型的性能。該策略包括數據增強、學習率策略和正則化技術等，有助於提高模型的魯棒性和泛化能力。

YOLOv5 在目標檢測方面具有較高的準確性和實時性能，使其在各種應用場景中得到廣泛應用，包括圖像分類、行人檢測、交通監控等。

2. Yolo v7

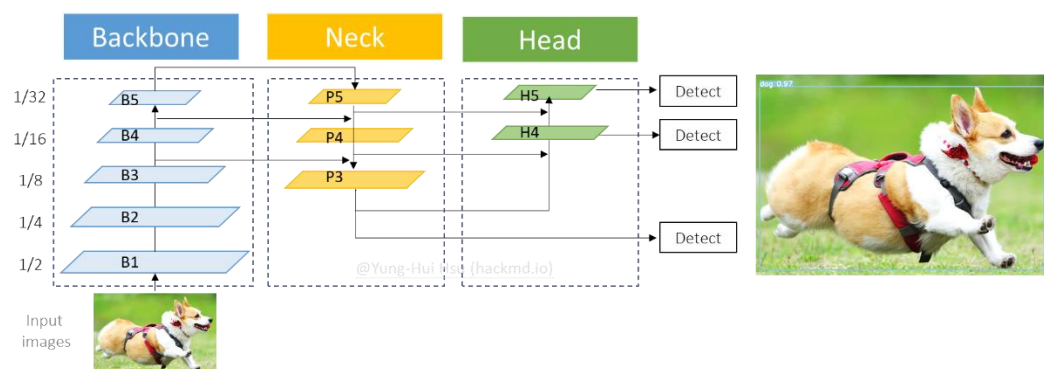


圖 2. yolo v7 架構簡圖

YOLOv7 通過引入多項架構優化，提升了速度和準確性。與 Scaled YOLOv4 類似，YOLOv7 的骨幹網絡不使用 ImageNet 預訓練的模型，而是完全使用 COCO 數據集進行訓練。這與作者在 Scaled YOLOv4 中的做法相似。

在 YOLOv7 的論文中，引入了以下主要變化：

1. 引入新的模型重參數化方法，使用無 identity 連接的 RepConv (RepConvN) 來設計網絡架構，以解決 RepConv 中的 identity 連接對於殘差和串聯結構的破壞問題。
2. 提出新的動態標籤分配策略，透過階層式的深層監督與動態標籤分配，提升特徵學習能力，稱為 coarse-to-fine lead guided label assignment。
3. 提出了有效利用參數和記憶體の"擴展"和"複合縮放"方法，以 ELAN 改進為基礎，提出 Extend-ELAN(E-ELAN)，在不破壞原始梯度路徑的情況下持續增強網絡學習的能力。
4. 有效地減少最先進的即時物件檢測模型的參數量和計算量，同時提高檢測精度和推理速度。

上述中的 1 和 2 被稱為"可訓練的贈品"(bag-of-freebies)，雖然增加了訓練成本，但能夠提高檢測的準確性，而且不會增加推理時的計算成本。

總的來說，YOLOv7 在 API 設計方面參考了 YOLOv5，並使用 PyTorch 進行了簡潔的實現。在模型架構設計和各種減少參數量、提升計算效能的改進方面，借鑒了近年來 CNN 神經網絡的各種性能優化技術。特別是在邊緣裝置運算方面，考慮到參數量和計算效率的進展。其中，Re-parameterized convolution 來自 RepVGG，簡化了 GoogleNet 的多分支概念，並在推理階段重現了單路徑無分支的 VGG 架構。E-ELAN 的多分支和 Group Convolution 則受到 AlexNet 的啟發，旨在分散運算。ELAN 來自 VovNet，而 VovNet 則源自 DenseNet 在計算量上的改進，提供了在邊緣裝置上使用的可行性。DenseNet 則繼承了 ResNet 的概念，通過特徵殘差取代輸入殘差。

在本次的研究中，我們 yolo v5 和 yolo v7 都進行實驗，發現 yolo v7 的辨識可以更加精細，並且準確度更高，因此這次的研究主要以 yolo v7 為主要模型。

二、資料前處理

根據圖三的資料分布圖，我們可以清楚地觀察到該資料集呈現極高的資料不平衡程度。這意味著不同類別的資料在資料集中的數量存在明顯的差異。面對這樣的情況，為了確保訓練模型的效果，我們需要進行相當高程度的資料前處理。

該資料集涵蓋了 32 個不同的類別，然而，有些類別的資料量相對較少。為了解決這個問題，我們採取了資料增量的方法，專注處理資料比例較少的類別。這些資料增量方法包括旋轉、隨機剪裁、調整亮度和尺寸等等。透過這些方法，我們能夠生成額外的訓練資料，增加資料集中這些少數類別的樣本數量。

具體而言，我們使用了旋轉操作來改變資料的方向和角度，隨機剪裁則能夠提供不同的圖像裁剪方式，調整亮度則能夠模擬不同光線條件下的圖像，而調整尺寸則能夠使圖像在不同比例下進行訓練。這些方法的實現方式和相關程式碼可以參考圖四所示。

透過這些資料增量方法，我們能夠為資料集中資料比例較少的類別提供更多樣本，從而平衡不同類別之間的資料分佈。這將有助於提高模型在訓練階段的效果和準確性，確保模型能夠適應各個類別的特徵和變異性。

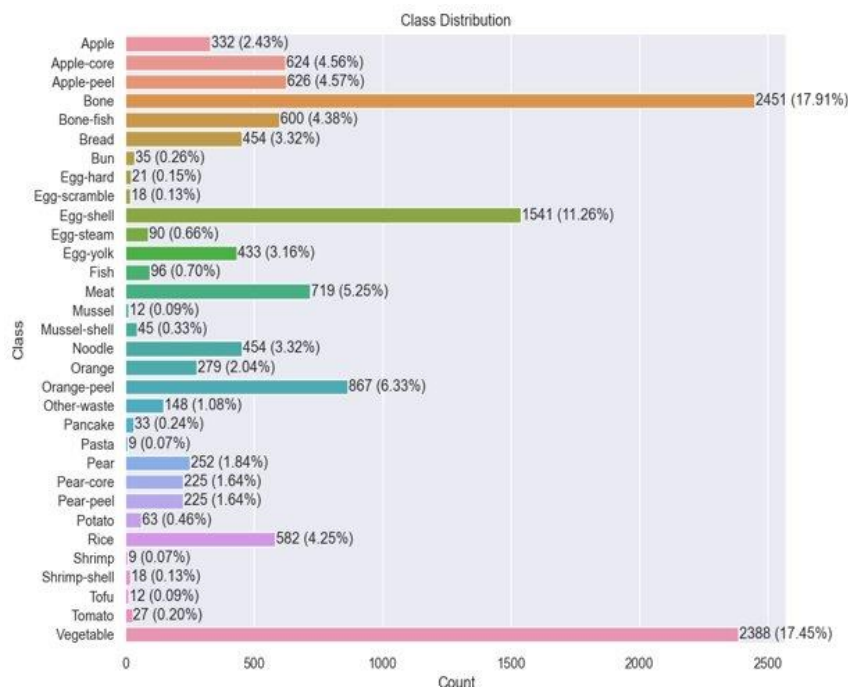


圖 3. 資料集標記分布圖

```
transform = transforms.Compose([
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomVerticalFlip(p=0.5),
    transforms.RandomRotation([90, 180, 270]),
    transforms.RandomResizedCrop(size=(224, 224), scale=(0.8, 1.2)),
    transforms.ColorJitter(brightness=0.15, saturation=0.15),
    transforms.Resize((640, 640)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

圖 4. 資料增量與前處理

三、模型建立與訓練

1. Yolo v5

首先我們先進行 yolo v5 的模型的實驗。經由上方研究方法的介紹，我們可以看到 yolo v5 的模型架構如下圖：

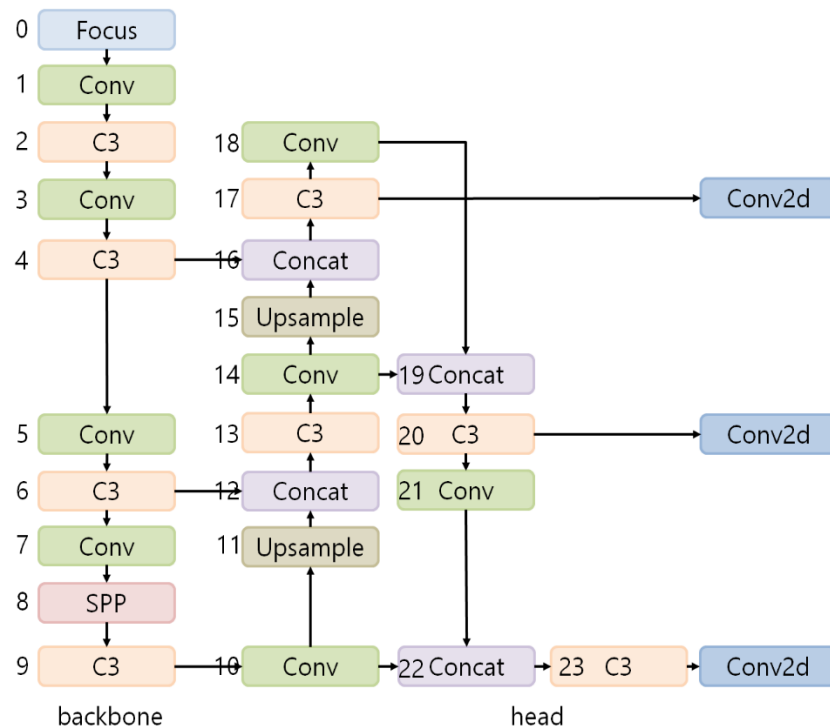


圖 5：網站圖_員工登錄後臺頁面

而我們實際的 model layer 結構圖則如下圖 6. 所示

	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	361984	models.common.C3	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90880	models.common.C3	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	296448	models.common.C3	[256, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C3	[512, 512, 1, False]
24	[17, 20, 23]	1	99789	models.yolo.Detect	[32, [[0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5]], [128, 256, 512]]

Model summary: 214 layers, 7105933 parameters, 7105933 gradients, 16.2 GFLOPs

圖 6：網站圖_員工登錄後臺頁面

在模型的超參數優化部分，我們針對批量大小（batch size）和迭代數（epoch）進行了優化，以提升模型的性能和效率。

首先，我們對不同的批量大小進行了評估，包括 32、64 和 128。我們分別計算了每個批量大小的 mAP_0.5、mAP_0.5:0.95 以及執行時間，以確定最佳的批量大小。

根據我們的實驗結果，批量大小為 64 的表現最佳。它在 mAP_0.5 和 mAP_0.5:0.95 方面均獲得了最高的分數，同時執行時間也最短。這表明使用批量大小為 64 可以獲得最佳的辨識性能和計算效率。

Batch size	mAP_0.5	mAP_0.5:0.95	Time (hrs.)
32	0.4152	0.2318	1.122
64	0.4199	0.2364	0.981
128	0.4181	0.2193	1.159

至於在迭代數（epoch）的部分，我們則是對於不同的迭代數進行實驗，包括 10, 30, 50，同時也分別計算了個別 mAP_0.5、mAP_0.5:0.95 以及執行時間，以確定最佳的迭代數。

Batch size	mAP_0.5	mAP_0.5:0.95	Time (hrs.)
10	0.2807	0.1466	0.367
30	0.4199	0.2364	0.981
50	0.4943	0.2773	1.85

經過上述的測試後，我們又進行了一次綜合性的 grid search 來確定最好的超參數組合

Epoch	Batch size	mAP_0.5	mAP_0.5:0.95	Time (hrs.)
10	32	0.299	0.1576	0.317
30	32	0.4152	0.2318	1.122
50	32	0.4799	0.2526	1.552
10	64	0.2807	0.1466	0.367
30	64	0.4199	0.2364	0.981
50	64	0.4943	0.2773	1.85
10	128	0.2703	0.1442	0.335
30	128	0.4181	0.2193	1.159
50	128	0.4981	0.2649	1.383

我們發現 epoch = 50 和 batch size = 128 的這個組合的結果又更好了。

但此時因為我們發現在 50 epoch 的時候會發生 overfitting 的情形，如下圖所示

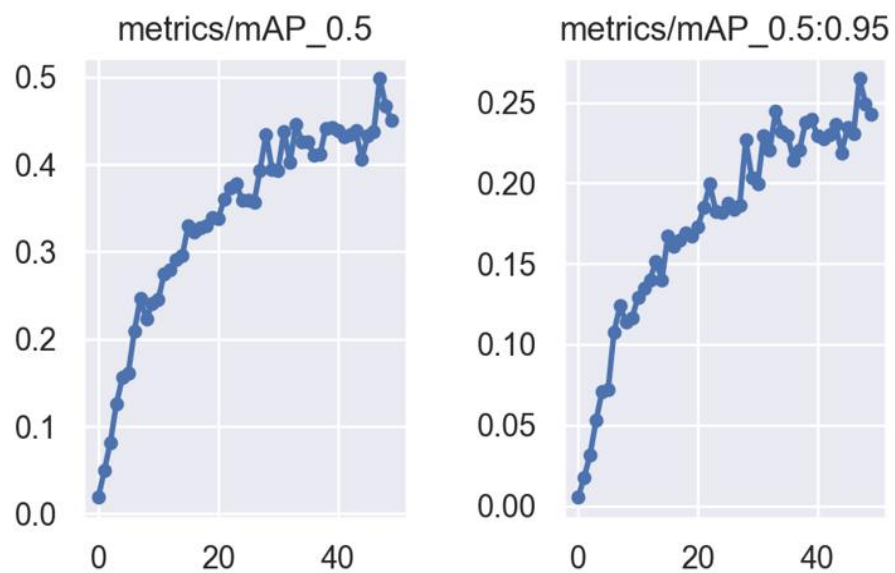


圖 7. mAP 測試結果

model loss

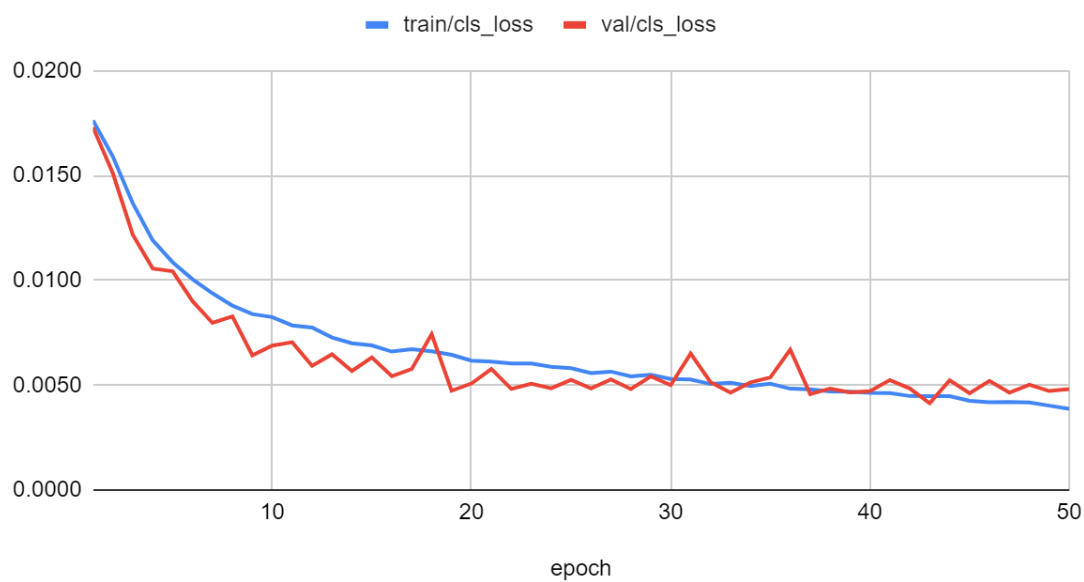


圖 8. loss

而使用 yolo v5 和以上的超參數組合訓練出來的模型的辨識結果如下兩圖所示

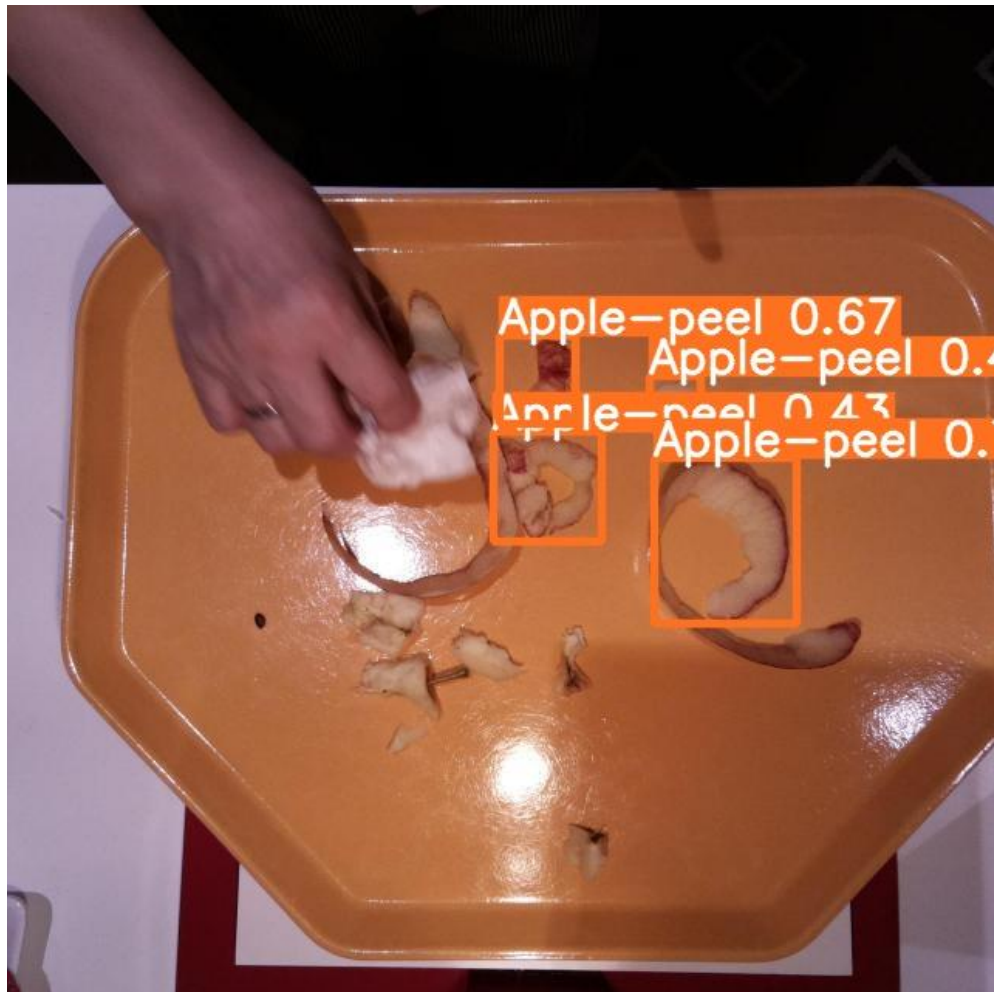


圖 9. Yolo v5 辨識結果 1



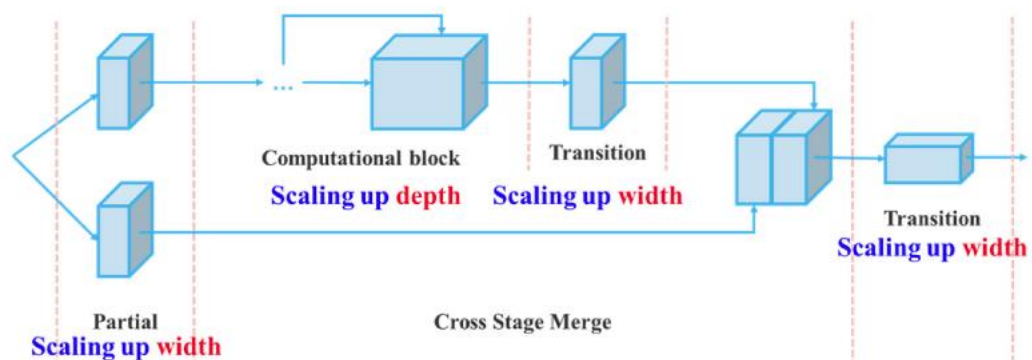
圖 10. Yolo v5 辨識結果 2



圖 11. Yolo v5 辨識結果 3

2. Yolo v7

接著我們先進行 yolo v7 的模型的實驗。經由上方研究方法的介紹，我們可以看到 yolo v7 的模型架構如下圖：



(c) compound scaling up depth and width for concatenation-based model

圖 12. Yolo v7 模型圖 1

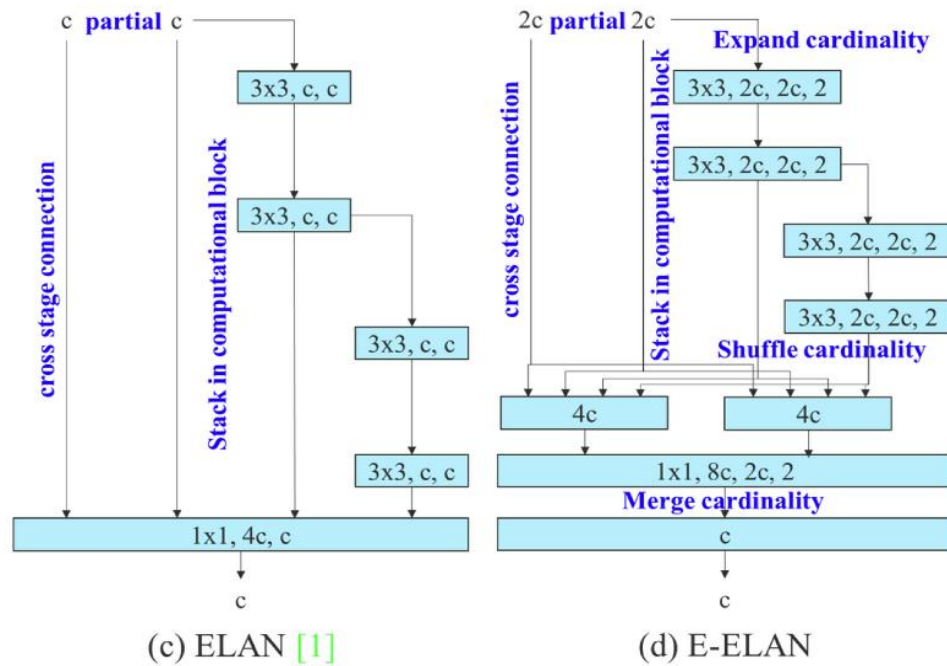


圖 13. Yolo v7 模型圖 2

而我們實際的 model layer 結構圖則如下圖. 所示

81	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
82	-2	1	131584	models.common.Conv	[512, 256, 1, 1]
83	-1	1	295168	models.common.Conv	[256, 128, 3, 1]
84	-1	1	147712	models.common.Conv	[128, 128, 3, 1]
85	1	1	147712	models.common.Conv	[128, 128, 3, 1]
86	-1	1	147712	models.common.Conv	[128, 128, 3, 1]
87	[-1, -2, -3, -4, -5, -6]	1	0	models.common.Concat	[1]
88	-1	1	262656	models.common.Conv	[1024, 256, 1, 1]
89	1	1	0	models.common.MP	[1]
90	-1	1	66048	models.common.Conv	[256, 256, 1, 1]
91	-3	1	66048	models.common.Conv	[256, 256, 1, 1]
92	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
93	[-1, -3, 51]	1	0	models.common.Concat	[1]
94	-1	1	525312	models.common.Conv	[1024, 512, 1, 1]
95	-2	1	525312	models.common.Conv	[1024, 512, 1, 1]
96	-1	1	1180160	models.common.Conv	[512, 256, 3, 1]
97	-1	1	590336	models.common.Conv	[256, 256, 3, 1]
98	-1	1	590336	models.common.Conv	[256, 256, 3, 1]
99	-1	1	590336	models.common.Conv	[256, 256, 3, 1]
100	[-1, -2, -3, -4, -5, -6]	1	0	models.common.Concat	[1]
101	-1	1	1649600	models.common.Conv	[2048, 512, 1, 1]
102	75	1	328704	models.common.RepConv	[128, 256, 3, 1]
103	88	1	1312768	models.common.RepConv	[256, 512, 3, 1]
104	101	1	5246976	models.common.RepConv	[512, 1024, 3, 1]
105	[102, 103, 104]	1	199245	models.yolo.Delegl	[32, [[12, 16, 19, 36, 40, 28], [36, 75, 76, 55, 72, 146], [142, 110, 192, 243, 459]]]

Model Summary: 407 layers, 37361645 parameters, 37361645 gradients, 105.6 GFLOPS

圖 14. Yolo v7 模型 layers

在模型的超參數優化部分，我們針對迭代數（epoch）和批量大小（batch size）進行了優化，以提升模型的性能和效率。

首先，我們對不同的迭代數進行了評估，包括 10, 30（因為硬體設備限制的關係所以沒辦法執行 50 代的運算）。我們分別計算了每個批量大小的 mAP_0.5、mAP_0.5:0.95 以及執行時間，以確定最佳的批量大小。

根據我們的實驗結果，迭代數為 30 的表現最佳。它在 mAP_0.5 和 mAP_0.5:0.95 方面均獲得了最高的分數，同時執行時間也最短。這表明使用迭代數 30 可以獲得最佳的辨識性能和計算效率。

Epoch	mAP_0.5	mAP_0.5:0.95	Time (hrs.)
10	0.3227	0.1856	0.466
30	0.5413	0.3364	1.44

至於在批量大小（batch size）的部分，我們則是對於不同的批量大小進行實驗，包括 8, 16，同時也分別計算了個別 mAP_0.5、mAP_0.5:0.95 以及執行時間，以確定最佳的迭代數。

根據我們的實驗結果，批量大小為 16 的表現最佳。它在 mAP_0.5 和 mAP_0.5:0.95 方面均獲得了最高的分數，同時執行時間也最短。這表明使用批量大小為 16 可以獲得最佳的辨識性能和計算效率。

Batch size	mAP_0.5	mAP_0.5:0.95	Time (hrs.)
8	0.4269	0.2116	0.543
16	0.5413	0.3364	1.44

使用 epoch = 30 和 batch size = 16 的超參數設置下進行辨識，
結果如下圖：



圖 15. Yolo v7 辨識結果

叁、網頁頁面與功能介紹

一、網頁管理者系統及其頁面介紹

1. 員工登錄頁面：利用員工 ID 及密碼登錄後臺管理系統

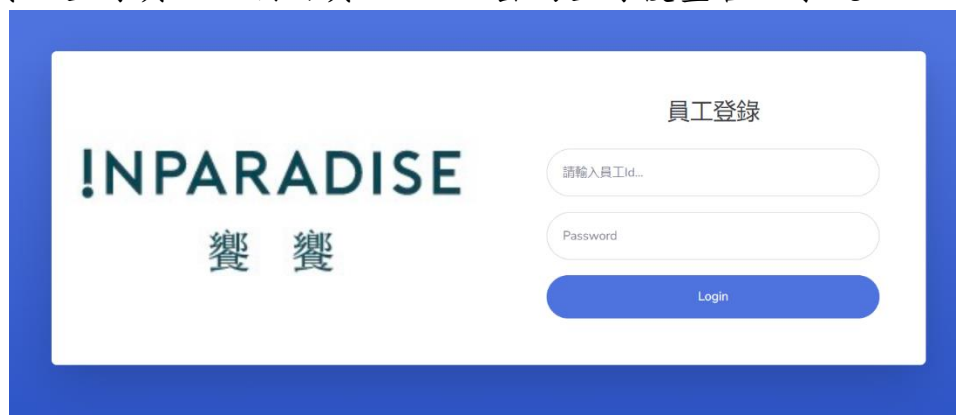


圖 16：網站圖_員工登錄後臺頁面

2. 後臺管路系統首頁：員工可在後臺管理系統觀察營業狀況及做會員或訂單等調整

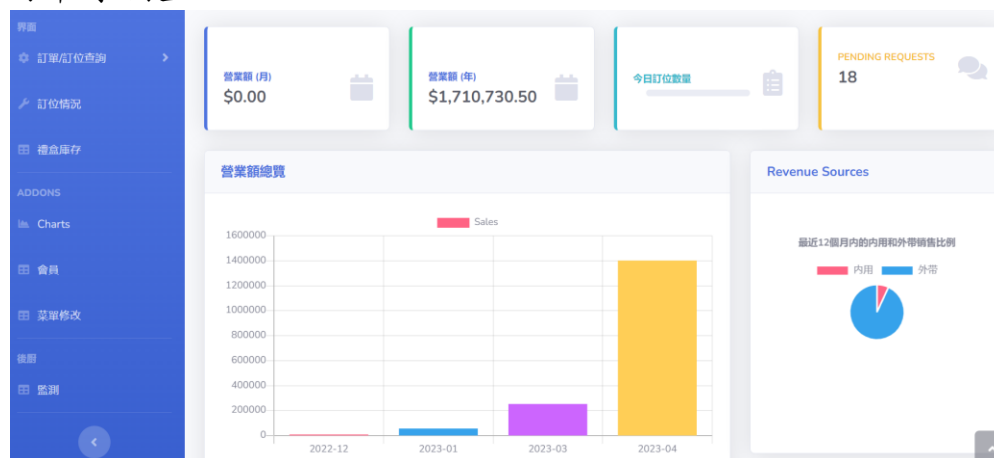


圖 17：網站圖_後臺管理系統首頁

- a. 儀表板：可得知月和年營業額及今日餐廳訂位數量



圖 18：網站圖_後臺管理系統儀表板

- b. 年度營業總覽：利用長條圖和圓餅圖得知每月營業額及營業額主要來源

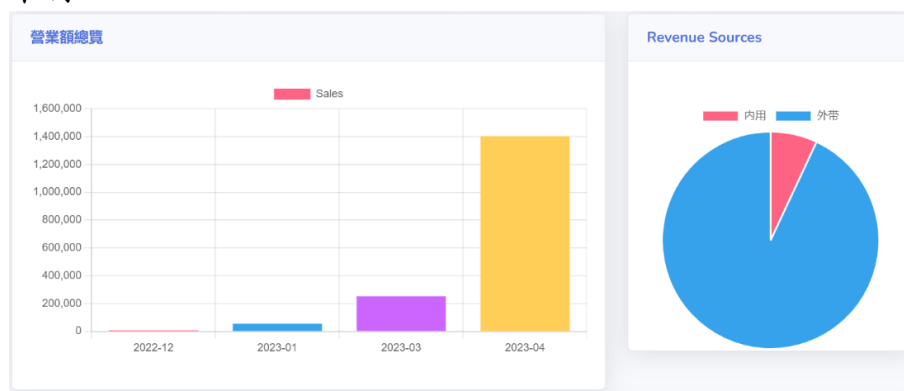


圖 19：網站圖_後臺年營業額圖表

- c. 本月餐廳訂位情況：可直接得知這個月每天訂位數量

本月訂位情況

日期格子內數字為當天已預定之位置總數

2023-04						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1 2
2	3	4	5	6	7	8
9 2	10	11	12	13 3	14	15
16	17	18	19	20 3	21	22
23	24	25	26	27	28	29
30						

圖 20：網站圖_本月訂位月曆

3. 訂單與訂位

a. 訂單查詢與修改：可以直觀查詢和修改訂單明細

訂單修改

訂單

Show 10 entries Search:

Order ID	Member ID	Member Name	Contact num	Products	Total Price	Pickup Date	Pickup Time
1	1	firstuser	123456789	遊歷餐餐組合_八品套餐 *1 Total_price: *2480	2480	2023-04-28	16:00:00
11	1	1	1	遊歷餐餐組合_八品套餐 *1 Total_price: *1680	1680	2023-04-26	06:15:00

Showing 1 to 2 of 2 entries

Previous 1 Next

圖 21：網站圖_訂單明細

*輸入者/用戶Id:

*名字:

電話:

遊歷餐餐組合_八品套餐:

獨餐旬寶盒_牛:

獨餐旬寶盒_豚:

京都醬燒鰻魚定食_附原味茶碗蒸:

炙烤牛小排定食_附原味茶碗蒸:

上選九品刺身盛合:

旬鮪刺身盛合:

旬鮪握壽司盛合_18貫:

極品握壽司盛合_10貫:

極選海陸匯_牛小排_干貝_草蝦:

黑松露牛小排_6個:

極選海陸匯_羊排_干貝_草蝦:

燒烤豬肋排_6支:

法式烤羊排_6支:

地中海香料海鮮:

XO醬燴雙鮮:

牛小排串燒:

姿燒香魚:

杏仁炸蝦天婦羅佐黃金甜醬:

伊比利豬肋條燒:

鮑魚海鮮粥:

石斑魚味噌湯:

中卷佐鳳梨莎莎:

燻鮭凱薩沙拉:

【甜田珠寶盒】:

【餐餐茗茶獻禮】:

*領取日期:

領取時間:

新增 重新填寫

圖 22：網站圖_新增訂單

*訂單Id:

*輸入者/用戶Id:

*名字:

電話:

遊歷餐餐組合_八品套餐:

獨餐旬寶盒_牛:

獨餐旬寶盒_豚:

京都醬燒鰻魚定食_附原味茶碗蒸:

炙烤牛小排定食_附原味茶碗蒸:

上選九品刺身盛合:

旬鮪刺身盛合:

旬鮪握壽司盛合_18貫:

極品握壽司盛合_10貫:

極選海陸匯_牛小排_干貝_草蝦:

黑松露牛小排_6個:

極選海陸匯_羊排_干貝_草蝦:

燒烤豬肋排_6支:

法式烤羊排_6支:

地中海香料海鮮:

XO醬燴雙鮮:

牛小排串燒:

姿燒香魚:

杏仁炸蝦天婦羅佐黃金甜醬:

伊比利豬肋條燒:

鮑魚海鮮粥:

石斑魚味噌湯:

中卷佐鳳梨莎莎:

燻鮭凱薩沙拉:

【甜田珠寶盒】:

【餐餐茗茶獻禮】:

*領取日期:

領取時間:

訂位修改 重新填寫

圖 24：網站圖_修改訂單

取消訂單

*訂單ID: (注意: 是訂單編號, 不是用戶id)

取消訂單

圖 23：網站圖_刪除訂單

b. 訂位查詢與修改：可以直觀查詢和修改訂位明細

訂位修改

訂位

Show entries Search:

Reserved_Id ↑↓	Name ↑↓	ID ↑↓	Seats ↑↓	Date ↑↓	Time ↑↓	Email ↑↓	Phone NO. ↑↓
666	member2	2	2	2023-04-23	19:00:00	member2@gmail.com	123456789
667	firstuser	1	9	2023-04-23	16:00:00	firstuser@gmail.com	333444

Showing 1 to 2 of 2 entries Previous **1** Next

圖 25：網站圖_訂位明細

新增訂位

*輸入者id:

*訂位者名字:

*訂位日期:

訂位時間:

預定數量:

電話:

E-mail 帳號:

新增 重新填寫

圖 26：網站圖_新增訂位

取消訂位

*訂位ID: (注意: 是訂位編號, 不是用戶id)

取消訂位

圖 27：網站圖_取消訂位

5. 禮盒庫存：可以讓員工得知目前禮品禮盒之庫存數量

禮品/禮盒庫存

Show entries

Search:

貨品編號	貨品名稱	庫存量
1	甜田珠寶盒	100
2	饕饕茗茶獻禮	200

Showing 1 to 2 of 2 entries

Previous **1** Next

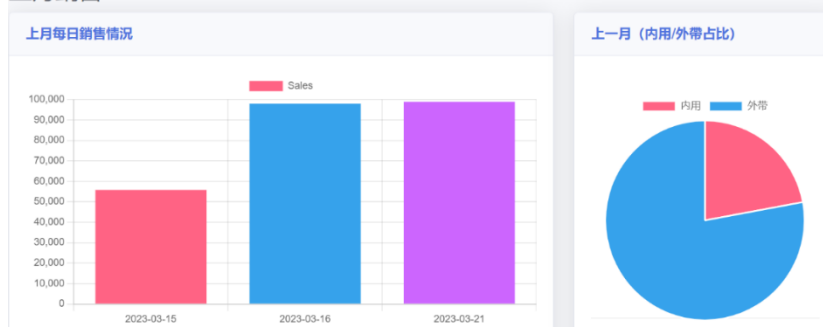
圖 30：網站圖_禮品禮盒庫存

6. 銷售情況：直觀觀察每月的營業狀況

本月銷售



上月銷售



近一年銷售

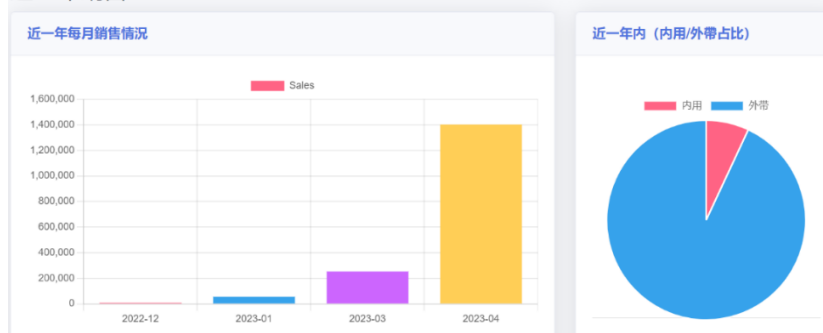


圖 31：網站圖_餐廳營業狀況

7. 會員管理系統

a. 會員總覽：可以直接搜索所有會員資料

響響會員用戶個人資料

所有登錄會員之資料，所有儲存於資料庫之會員用戶資料皆出現於此

會員資料

Show 10 entries Search:

Name	ID	Sex	Birth	Phone NO.	Email
member2	2	Male	2023-04-13	123456789	member2@gmail.com
third	1	男	2023-04-04	333	000.com
Name	ID	Sex	Birth	Phone NO.	Email

Showing 1 to 2 of 2 entries

Previous 1 Next

圖 32：網站圖_會員總覽

b. 新增會員資料

新增會員

*使用者帳號: (請使用英文或數字鍵)

*使用者密碼: (請使用英文或數字鍵)

*密碼確認: (再輸入一次密碼)

*性別: ☒ 男 ☐ 女

*生日: 年/月/日

電話:

E-mail 帳號:

加入會員 重新填寫

圖 33：網站圖_新增會員

c. 刪除會員資料

刪除會員

*ID:

刪除會員

圖 34：網站圖_刪除會員資料

d. 修改會員資料

會員資料修改

*ID:

*使用者帳號: (請使用英文或數字鍵)

*使用者密碼: (請使用英文或數字鍵)

*密碼確認: (再輸入一次密碼)

*性別: ☒ 男 ☐ 女

*生日: 年/月/日

電話:

E-mail 帳號:

會員修改 重新填寫

圖 35：網站圖_修改會員資料

8. 菜單管理系統

a. 菜單總覽：可以查看現有的所有菜單

Show 10 entries Search:

Food ID	Food Name	category	introduce	Price	Food_picture	display
1	母親節海陸饗宴	母親節特備	母親節特備	1680	images/demo/gallery/2.jpg	0
2	桃山鰻魚丼	丼丼	桃山精選鰻魚佐特調醬汁	870	images/demo/gallery/3.jpeg	1
3	尊爵_海上盛宴	海上聖宴	尊貴豪華海鮮聖餐	780	images/demo/gallery/4.jpg	1
4	鐵板天使紅蝦	海上聖宴	鐵板炙燒老虎蝦	580	images/demo/gallery/5.jpg	1
5	廚神紅酒燴牛肉	主廚特選肉肉	主廚精選紅酒燴日本和牛	680	images/demo/gallery/6.jpg	1
6	德國豬腳	主廚特選肉肉	德國空運飼養豬	380	images/demo/gallery/7.jpg	1
7	極上_海上皇宮	海上聖宴	海龍宮盛宴	1240	images/demo/gallery/8.jpg	1
8	哲魯蛋糕	母親節特備	母親節限定孝親蛋糕	980	images/demo/gallery/9.jpg	0

Showing 1 to 8 of 8 entries Previous 1 Next

圖 36：網站圖_菜單總覽

b. 新增菜單資料

選擇檔案 未選擇任何檔案 Upload

*新菜色名字:

*category:

價格:

照片路徑: (/*/foodpic.jpg)

簡介:

是否顯示: ☒ yes ☐ no

新增 重新填寫

圖 36：網站圖_新增菜單資料

c. 修改菜單資料

*Food_Id:		
菜色名字:		(可填新名)
category:		
價錢:		
照片路徑:		(/*/foodpic.jpg)
簡介:	0	
是否顯示:	☒ yes ☐ no	
Menu修改 重新填寫		

圖 37：網站圖_修改菜單資料

d. 修改後前台菜單



圖 38：網站圖_修改後前台菜單資料

e. 修改後前台訂單表單

自取日期:

年 / 月 / 日

自取時間:

-- :-- :--

桃山鰻魚丼 價格: 870 請選擇您要的數量: 0	尊爵_海上盛宴 價格: 780 請選擇您要的數量: 0
鐵板天使紅蝦 價格: 580 請選擇您要的數量: 0	廚神紅酒燴牛肉 價格: 680 請選擇您要的數量: 0
德國豬腳 價格: 380 請選擇您要的數量: 0	極上_海上皇宮 價格: 1240 請選擇您要的數量: 0

送出訂單 取消訂單

圖 38：網站圖_修改後前台訂單表單

f. 後台訂單表單

*訂單Id: *輸入者/用戶Id:

*名字: 電話:

*領取日期: 年 / 月 / 日 ☐ 領取時間: -- :-- :--

桃山鰻魚丼 價格: 870 請選擇您要的數量: 0	尊爵_海上盛宴 價格: 780 請選擇您要的數量: 0
鐵板天使紅蝦 價格: 580 請選擇您要的數量: 0	廚神紅酒燴牛肉 價格: 680 請選擇您要的數量: 0
德國豬腳 價格: 380 請選擇您要的數量: 0	極上_海上皇宮 價格: 1240 請選擇您要的數量: 0

圖 39：網站圖_後台訂單表單

菜單與訂單表單會隨著各自得 display 而選擇顯示或隱藏。

9. 廚餘 A I 分析與統計系統

- a. 顧客空餐盤辨識偵測區
如果顧客用餐區上方的相機偵測到顧客的餐盤中並無物體，則代表顧客已用餐完畢，可以進行餐桌整理



圖 40：網站圖_顧客空餐盤辨識偵測區

- b. 廚餘區圖片物件辨識區
此處上傳餐盤中的廚餘部分，便可以辨識廚餘的物件，並且將偵測到的物件內容進行統計，以利廚師可以根據資料調整餐點口味

檢測結果情況

輸入圖片地址：

Al/image/test/8.jpg

开始延迟

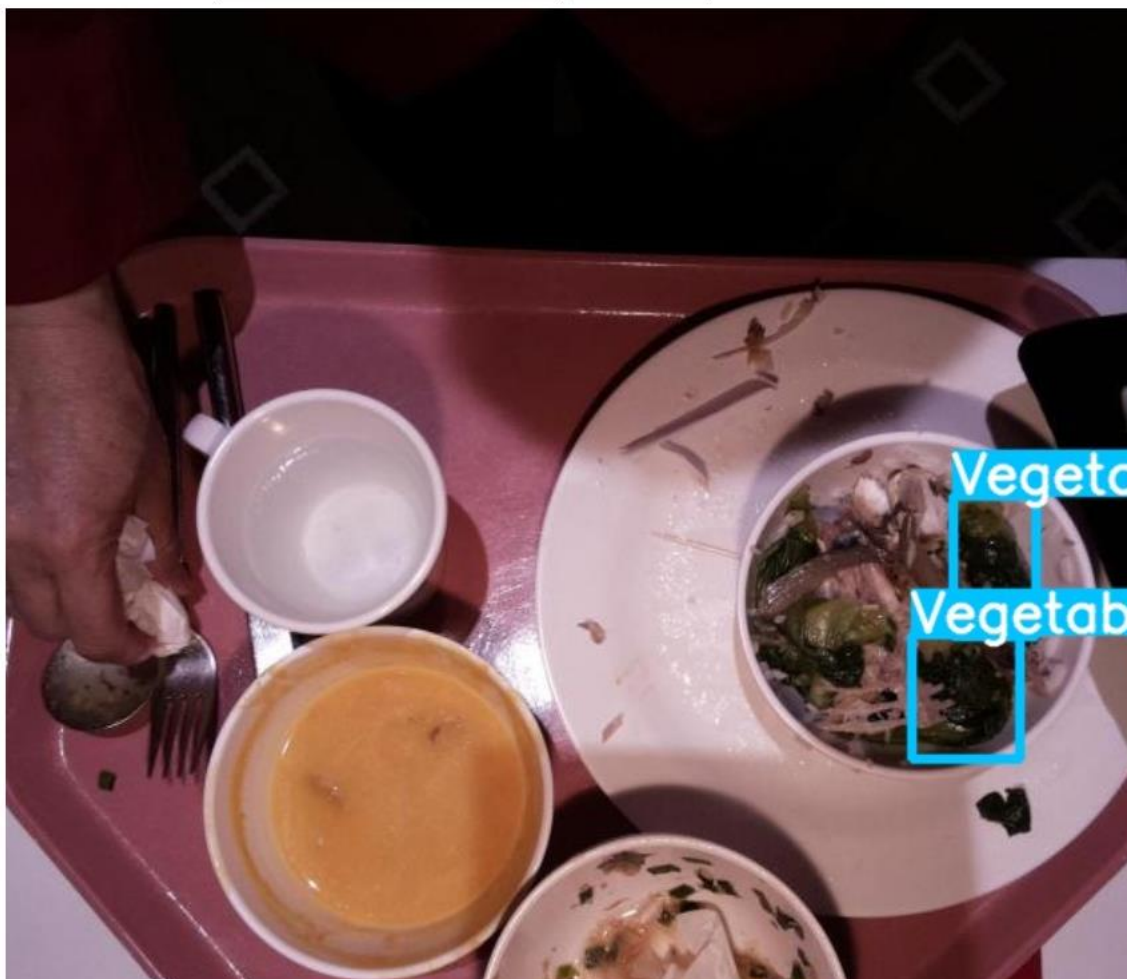


圖 41：網站圖_廚餘區圖片物件辨識區

c. 廚餘資料統計區

此處可以將已辨識到的廚餘內容資料進行整合至一個 CSV

厨余

選擇檔案

沒有選擇檔案

上传csv

圖 42：網站圖_廚餘資料統計區

肆、結論

在本研究中，我們致力於開發一個辨識吃到飽餐廳中客人所使用完畢的空餐盤的系統，並同時辨識出空餐盤中的廚餘類別。經過實驗和測試，我們得出了以下結論和研究成果的總結：

1. 資料收集和標註：我們成功收集了吃到飽餐廳中客人使用完畢的空餐盤的影像數據，並對這些影像進行了標註，標明了餐盤的位置和廚餘類別。這為我們後續的研究提供了寶貴的資料基礎。
2. 影像辨識模型：我們選擇了使用 YOLOv5 和 YOLOv7 兩種影像辨識模型進行實驗。這兩種模型都展現出了良好的性能，能夠準確地辨識出空餐盤的位置和廚餘類別。其中，YOLOv7 在速度和準確性方面取得了顯著的改進，相比於 YOLOv5，它減少了參數量和計算量，同時具有更快的推理速度和更高的檢測精度。
3. 資料前處理和增強：由於資料集中存在著極高的資料不平衡程度，我們進行了相當高程度的資料前處理。透過旋轉、隨機剪裁、調整亮度和尺寸等資料增強方法，我們成功地平衡了資料集中不同類別的資料分佈，提高了模型的訓練效果和準確性。
4. 系統性能評估：我們對開發的系統進行了全面的性能評估。結果顯示，我們的系統能夠準確地辨識出吃到飽餐廳中客人使用完畢的空餐盤，並且能夠準確識別出其中的廚餘類別。這對於服務人員收集空餐盤和廚師們及時調整食物口味具有重要意義。

總的來說，本研究的結果表明，我們開發的辨識空餐盤和廚餘類別的系統具有良好的性能和準確性。這對於提高吃到飽餐廳的運營效率、減少食物浪費和促進廚餘管理具有重要的應用價值。

然而，仍有一些潛在的研究方向值得探索和深入研究，以進一步改進系統的性能和擴展其應用範圍。以下是未來的研究方向的討論：

1. 擴大資料集：目前我們使用的資料集主要來自吃到飽餐廳，但可以進一步擴大資料集的規模和多樣性。收集更多來自不同類型餐廳的資料，並且考慮更多不同環境條件下的變化，如光照、背景等，以提高系統的魯棒性和泛化能力。

2. 廚餘分類和量化：除了辨識廚餘類別外，進一步研究和發展更精確的廚餘分類方法。探索基於影像和數據結合的技術，例如使用傳感器或智能設備進行重量和成分的量化，從而實現更精確的廚餘分析和管理的。
3. 系統優化和效能提升：繼續改進系統的效能和速度，以實現更高效的實時辨識和處理。探索硬件加速、模型壓縮和優化等方法，以提高系統的運行效率和節能性。
4. 餐廳管理整合：將該系統與餐廳管理系統進行整合，實現自動化的餐盤收集和廚餘管理。與餐廳的POS系統和訂單管理系統相連接，實現餐盤識別和廚餘分類的自動化，從而提高餐廳運營效率和資源管理。

總而言之，本研究通過引入先進的影像辨識技術，成功開發了一個能夠準確辨識空餐盤並識別廚餘的系統，具有提升餐廳運營效率、減少食物浪費、促進資源利用和環境保護等實際應用價值。