

國立台北大學資訊工程學系專題報告

Deep-Learning-based Mobile Application for Estimating Restaurant's Waiting Time

基於深度學習之預估餐廳等待時間的手機應用程式開發

專題組員：黃弘諺、劉俐妤、李華涓、楊喻丞

專題編號：PRJ-NTPUCSIE-111-009

執行期間：2022 年 9 月至 2023 年 6 月

1、摘要

隨著智慧時代的來臨，人工智慧與大數據等先端資訊技術紛紛導入餐飲業。為提供高水準的顧客服務與降低人工管理成本，全球的餐飲業者對於「智慧餐廳」的期待也逐漸提升。我們觀察到餐廳的候位流程，通常是利用人力的方式，以不精準的方式預估候位的時程，而不準確的候位預估時間通常讓店家與顧客浪費許多時間成本在等待，甚至進一步造成顧客的不滿，導致餐廳的評價受到影響。本計畫預計製作一款即時判斷食物剩餘量並估算出餐桌剩餘使用時間的應用軟體，店家透過深度攝影機接收到食物資訊，採用 YOLOX 作為目標檢測的模型，再透過 SegFormer 語意分割模型將目標分割。為了提高精準度，我們將深度攝影機的數據與分割後的目標合併計算。最後，將剩餘量輸入至回歸模型(Linear Regression Model)中來預測最終的剩餘時間，不同類型食物會有相對應的真實剩餘量曲線以及剩餘時間曲線，使用時會根據當前的曲線找到曾經學習過最相近的曲線來預測當前餐桌剩餘使用時間。而此系統可以於解決上述候位等實際問題。

2、研究動機與研究問題

如今，全球的餐飲業已開始導入人工智慧、物聯網等軟硬體，藉以達到餐廳年度營運目標與提升消費者體驗，綜觀目前已發展的餐飲資訊服務，仍缺乏預估何時可入座餐廳的系統，當顧客抵達餐廳或提早透過電話詢問餐廳的等候

時間，若餐廳為客滿狀態，往往都需要再花費寶貴時間等待入座。除此之外，當客滿的時候詢問店家預計要等待的時間，店家給出的答案往往模稜兩可或不準確，缺乏客觀且有系統的計算數據，若在餐廳附近走走逛逛，在離開的幾分鐘內餐廳可能就有位置了，又或是只須等個幾分鐘的位置，卻在旁站了快一小時。

有鑑於此，本計畫想結合人工智慧與深度攝影機，開發一款智慧餐廳系統，除了方便顧客查看餐廳的等候時間，也方便餐廳服務生掌握餐廳內每張餐桌的用餐狀況。

3、研究方法與流程

我們預計設計及開發一套人工智慧系統來檢測餐點剩餘量以及剩餘食用時間，首先每桌由個別的深度攝影機 (Depth Camera) 取得 RGB 圖像資訊，交由目標檢測模型(YOLOX)分別框出桌上所有的目標餐點，再將框出的圖片分別交由分割模型(SegFormer)準確標出食物所佔的像素點，結合深度攝影機的深度資訊即可得出剩餘量。接著將餐點剩餘量送入傳統回歸模型(Regression Model)中預估剩餘時間。然後藉由一款 APP 滿足店家與客人的需求，在註冊時可以透過選擇店家或是客人，相對應地，會有不同的操作介面。在店家介面下，店家能即時掌控各桌情況，以及調整模型參數。在客人介面下，客人點開 APP 就能了解店家的情況，如是否客滿、大概須

等多久才有位置。本系統的流程架構如圖 1 所示。

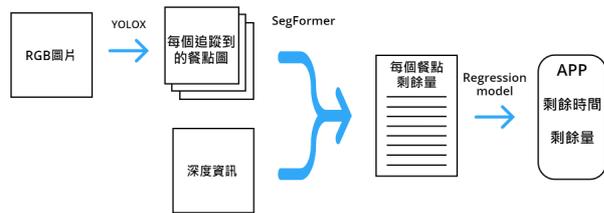


圖 1：預估候位時間系統流程架構圖



圖 2：目標檢測框類別與追蹤 id 圖

1. 人工智慧系統

I. 目標檢測

本計畫的做法如下，首先將深度攝影機連上電腦，取得 RGB 與深度資訊。由於不同食物之間的樣貌以及進食所需時間差異大，所以我們會根據不同種類食物進行預測。這裡我們使用 YOLOX 作為目標檢測模型，此模型主要作用為「框選出指定食物」以及「食用過程中追蹤食物」。

框選出指定食物：我們訓練的目標檢測模型可以框選出 9 種不同類別的食物。

食用過程中追蹤食物：由於系統需要運行三種模型，在系統上無法加載巨型深度學習模型，所以在食用過程中，食物追蹤的部分不使用追蹤模型，而使用相對直觀的方法進行算法追蹤。主要是透過前後兩幀畫面上，利用食物圖像是否有重疊，來判定是否為同樣食物，當兩個標記框重疊率超過指定閾值時，我們認定可能為相同食物，接下來會檢查前後兩幀目標檢測的分類是否為同一類，當以上兩種情況都滿足時，則我們認定為相同食物，藉由此方法進行食用過程中的食物追蹤。

II. 分割模型與深度資訊

將目標檢測框出來的圖片分割出來後送入分割模型(SegFormer)，找出食物所佔的像素點，對於不同類別的食物我們會使用不同權重的分割模型來進行跟分割檢測，也就是我們會載入 9 個不同權重的分割網路。餐點食用剩餘量主要是透過體積來進行預測，主要分成「初始體積」以及中間「食物體積變化」。

初始體積：店家可透過設定軟體的設定初始深度按鈕記錄一張初始深度圖，按下按鈕後紀錄 90 幀的深度資訊，對 90 幀的每個像素點取平均，會得到一張平均後的深度圖，並將該圖以 numpy 矩陣格式儲存。

食物體積變化：該預估指在進食過程中，針對食物在食用過程中，一直取得分割以及深度資訊，將深度資訊與初始深度相減可得高度落差，將屬於食物區塊的高度落差加總可得食物體積，最後透過體積變化量計算出剩餘量，當最終食物吃完時，分割網路終將沒有該食物的像素點，所以「最終體積」將會是 0，如此代表食用完畢。



圖 3：分割網路結果

(藍色：BackGround、綠色：NotFood、紅色：Food¹)

我們進行圖 3 之解釋，說明如下：物體交界處判定的較為不準確，是因為在標記訓練資料圖片時，難免會有飯粒或食物的殘渣在碗壁上，不一定會有明顯的分割界線，導致模型在判定 NotFood 時較不精準，但模型用於預估剩餘量並未用到 NotFood 的資訊，所以並無影響。

III. 回歸模型

最後須透過迴歸模型預測出剩餘食用時間，由於不同種類食物所需花費的時間落差很大，某些食物可能約 10 幾分鐘就會吃完，某些食物甚至可能需要到 1 小時才會食用完畢，為了可以增加預測精確度，我們會根據不同類別的食物，在其上使用專屬的訓練權重，以進行個別食物的時間預測。

我們透過上面的剩餘量來推算出剩餘食用時間，將 t 秒內的剩餘量與經過的時間，即長度為 $t+1$ 的向量傳入回歸模型，即可得到剩餘時間。

¹ Food 為食物的像素點，NotFood 為碗的像素點，BackGround 為背景

2. 資料集

I. 訓練資料集

訓練資料集分三個部分，分別是目標檢測訓練集、語意分割訓練集、回歸模型訓練集。由於經過在網路常閱資料庫中搜尋，甚少有食物被食用的照片集，因此未能在網路上找到相應的資料集以滿足本計畫的訓練需求，因此使用的資料集均為人工產生，只有部分圖片透過網路下載。圖像來源於網路的公開資料集(Kaggle)、手動拍攝及透過 Google 表單蒐集民眾分享的圖片。

目標檢測訓練集：「目標檢測」使用標籤工具(Labelimg)手動生成 Yolo 邊界框(Bounding Box)，將目標食物包含容器用矩框標註出來(圖 4)，總共分為九個類別：Donburi, SoupRice, Rice, Countable, SoupNoodle, Noodle, Sidedish, SolidSoup, Soup²。



圖 4：餐點與容器的邊界框

語意分割訓練集：「語意分割」使用標註工具(Labelme)手動進行食物面積標註，分別將餐點中食物與容器的像素標記出來(圖 5)，分割總共分為兩個類別：食物與非食物。

² 九個類別分別為：丼飯、有湯的飯(例如稀飯)、飯(炒飯類)、可數的類別(例如水餃類)、湯麵、麵(乾麵)、小菜、有東西的湯、湯(清湯類)



圖 5：食物與非食物像素框

回歸模型訓練集：用深度攝影機搭配碼表錄製影片，紀錄每幀的深度資訊、RGB 圖、與碼表經過的時間。由深度資訊與 RGB 圖經過目標檢測與分割模型可得預測剩餘量，將預測剩餘量與經過的時間存入 excel 表中，即為回歸模型之訓練集。在訓練時將訓練集的每筆資料以每秒，將前 t 個剩餘量與當前經過時間作為一筆訓練資料，並將總長扣除當前經過的時間為標注資料。即有一筆資料為 $[3,2,1,0]$ ，令 t 為 3 則會產生兩筆訓練資料 $[3,2,1,3]$ 、 $[2,1,0,4]$ ，其標注分別為 1、0。

II. 驗證資料集

資料驗證分兩個部分，第一部分為檢測剩餘量，如圖 7，我們將整個吃飯過程的 RGB 與深度資訊保存下來，並在下方用秤重機，秤重的資訊可精確得知剩餘量。驗證時將保存的 RGB 以及深度資訊放入模型進行預測，所得結果與秤重機結果評估誤差，依照不同的剩餘量劃分區段，分別計算 L1 損失。第二部分為剩餘時間預估，如圖 8，依照所得影片長度即可得知準確之剩餘時間，依照不同的剩餘時間劃分區段，分別計算 L1 損失。

系統可行性驗證-剩餘量的平均誤差百分比

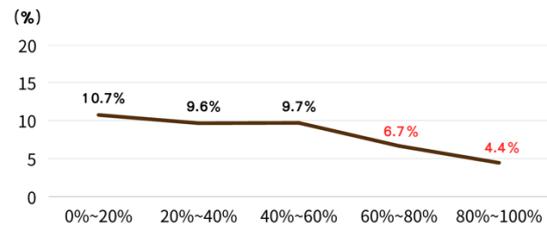


圖 7：剩餘量的平均誤差百分比

我們進行圖 7 之解釋，說明如下：經整理目前有的 63 筆資料，橫軸為時間經過的百分比，縱軸為剩餘量誤差的百分比。

系統可行性驗證-剩餘時間的平均誤差

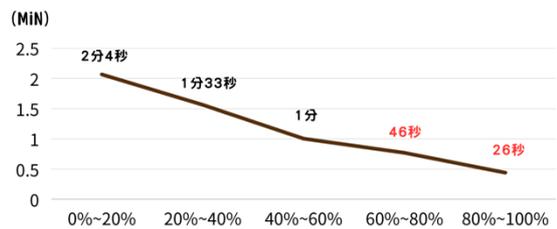


圖 8：剩餘時間平均誤差

我們進行圖 8 之解釋，說明如下：經整理目前有的 9 筆資料，橫軸為時間經過的百分比，縱軸為剩餘時間的誤差分鐘。

3. APP

I. App IDE 及程式語言的選用

目前市面上流行許多不同的整合開發環境(Integrated Development Environment, IDE)，例如 Android Studio、Eclipse、XCode 等等，其中 XCode 是目前 IOS 移動應用程式開發最受歡迎也是最流行的 IOS IDE，此我們選擇使用 XCode 作為系統應用程式的開發環境，並搭配「Swift」作為程式語言來開發 IOS 應用程式。

Swift 有著強大的開發環境、集合 Objective-C、Python、Ruby 與 JavaScript 等特性與優點，以簡單直覺的程式碼來實現原本較為複雜的程式碼，來實現互通性強大的功能，另外，

透過 Playground 也能更容易的預覽程式碼片段的執行結果，更容易的分析以及觀看目前程式碼的運行是否正常與如預期，因此我們選擇 Swift 作為 App 的程式語言。

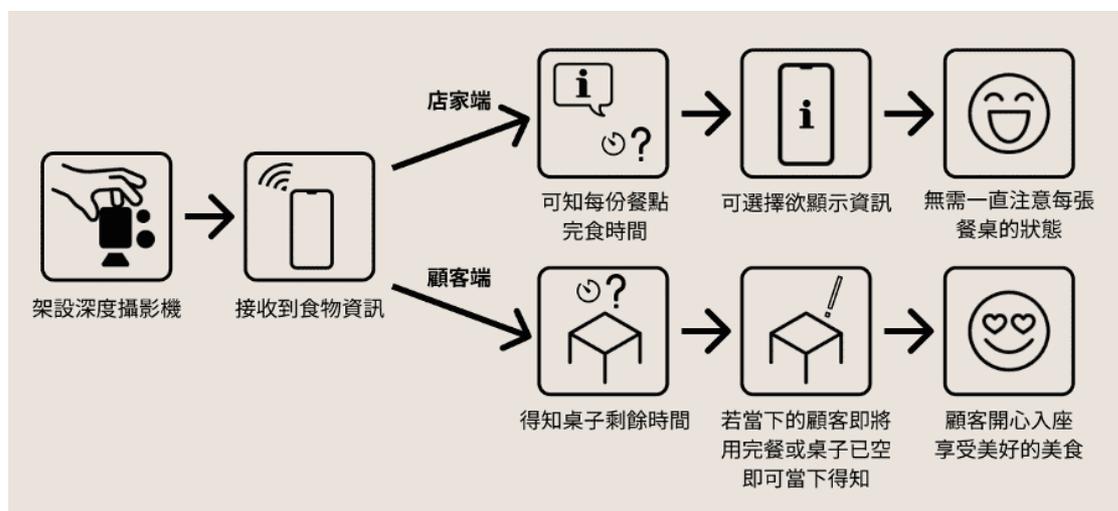


圖 9：APP 流程圖

II. APP 操作流程圖

我們最後的成果會呈現在開發的 APP 上供使用者使用，目前預計將 APP 設計為兩種不同使用者的操作介面，分別為店家端以及顧客端。店家端會收到每份餐點的剩餘量與預估剩餘時間的數據，也可自由操作介面開啟特定食物資訊的功能供顧客端查看；顧客端只能看到餐廳中每張桌子預計需等待之時間以及店家端所開啟讓顧客能掌握的資訊，如果該店家已出現空座位，APP 的畫面也會同步顯示。為了避免使用者因為對 APP 的操作流程不熟悉而感到困擾，我們製作了一個簡易的 APP 流程圖(圖 9)供參考，希望能讓使用者有更好的使用體驗。

III. APP 操作介面

A. 店家端

進入店家端的 APP 頁面之後(圖 10)，可以看見餐廳內深度攝影機所照射到的每份食物資訊，詳細的食物資訊如圖 11，店家可以獲得三種食物資訊，分別為食物編號、預估食物完食剩餘時間以及預估完食剩餘量，以方便店家預估翻桌的時間。在 APP 的右上方有快捷鍵可以選取，「資訊」功能可以獲得食物的線性圖與數值，其中數值包含了食物類別與剩餘量(圖 12)，「設定」功能提供店家在權重、空間與顧客端的顯示做設定，其中「權重設定」能讓店家選擇我們事先預訓練好的權重(圖 13)，另外也有比較的功能，可以選擇最適合的權重來判斷該店家的食物；「使用端設定」能讓店家選擇欲讓顧客端可見的資訊(圖 14)，顧客端打開 APP 頁面的時候，主頁就會顯示店家有開啟的資訊。



圖 10：店家端 APP 介面介紹

- (左 1)APP 初始畫面
- (左 2)新帳號建立端口
- (右 2)店家端接收到食物之資訊
- (右 1)右上方的功能選擇

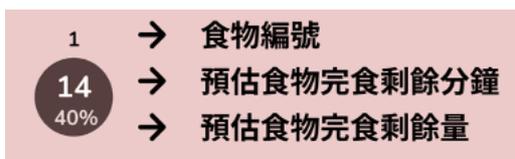
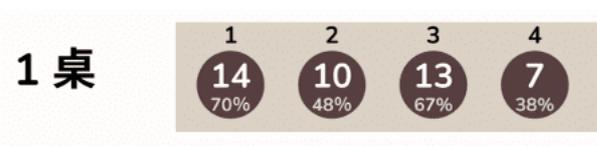


圖 11：詳細食物資訊

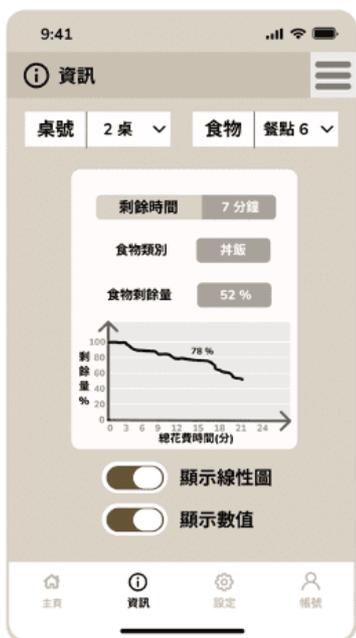


圖 12：食物資訊頁面



圖 13：權重設定



圖 14：使用端設定

B. 顧客端

進入顧客端的 APP 頁面之後，如圖 15，主頁會顯示店家所開啟的資訊，另外也能選擇只顯示該店家有空桌的畫面，畫面簡化流利，讓顧客能透過餐廳剩餘時間與空桌的狀態來做後續行程的判斷，也能提早預估餐廳能入座的時間，提供便捷的功能。

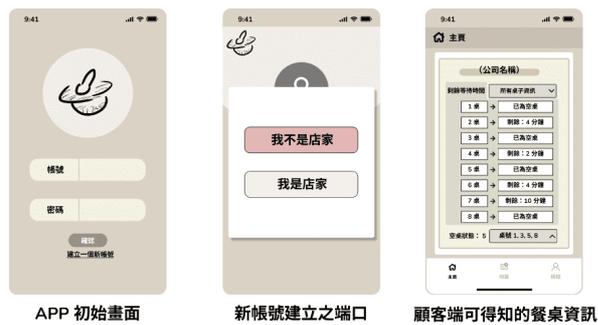


圖 15：顧客端 APP 介面介紹

(左)APP 初始畫面

(中)新帳號建立之端口

(右)顧客端可得知的餐桌資訊

4、 預期結果

目前有市面上流通或正在被開發與研究的餐飲業 AI 人工智慧機器學習大部分與機器人送餐、餐點製作、收付結帳有關，但在即時判斷餐桌狀態的相關研究可說是極度缺乏，餐廳內用的餐點吃完後、收拾完餐桌，下一組客人才能入座，若餐廳為客滿的狀態，顧客也無法得知需等候入座的確切時間，因此可能會影響到顧客後續時間規劃的問題。另外，餐桌若需要人工收拾餐盤，餐廳員工需要時常在餐廳內巡視。為了解決上述問題，本研究提出了一個透過機器學習即時預估餐桌剩餘使用時間的創新系統。

此研究的預期結果為藉由深度攝影機所拍攝的即時影像，根據機器學習的運算預估出每份餐點的剩餘時間，透過 APP，可得知每張餐桌的即時剩餘使用時間，方便顧客與店家參考與後續應用。

為了確保可順利完成本研究預期的進度，我們將進度大致分為前期的 AI 模型撰寫及食物資料搜集、中期硬體操作與資料驗證以及後期的撰寫 APP 及論文製作三個階段，以協助追蹤研究進度。

5、 需要指導教授指導內容

本計畫涉及的機器學習為指導教授的研究領域。在專題研究的過程中，實作各種模型與面對諸多限制與困難時，指導教授可以從旁給予建議；硬體操作方面，在做硬體選擇與記憶體不足需壓縮等方面，也需要與指導教授討論如何以何種方式解決問題；軟體設計方面，除了仰賴自己研究學習的相關知識外，也需要與指導教授討論實作的原理與內容。製作專題的期間也需要定期與指導教授討論研究內容，穩定推動專題前進同時也能避免研究方向的偏離。

綜上所述，透過指導教授用心、細心與耐心的指導，將能使研究更快又更順利的完成。

6、 參考文獻

- [1] [Online] <https://notes.andywu.tw/2020/>採用 yolov3 模型的自助餐菜色辨識結帳系統/
- [2] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun. YOLOX: Exceeding YOLO Series in 2021. arXiv, 2021.
- [3] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. arXiv, 2021.