愛文芒果—利用影像進行等級分類

黃柏叡 黃啟宏 潘躍升
r10725003@ntu.edu.tw
r10725028@ntu.edu.tw
r10725050@ntu.edu.tw

1. Introduction

愛文芒果為台灣重要出口農產品之一。愛文芒果得銷量於近幾年持續增長，不僅躍升為三大外銷高經濟生鮮果品之一，更將外銷國拓展至日本、中國、美國以及香港等地。雖然，在各國當地政府的政策配合下，台灣芒果比起以往提高了知名度並增加了市佔率，卻仍然遭遇了其他同為芒果出口國的削價競爭（譬如菲律賓、泰國等國）。因此，諸多品種改良、採收後的處理技術以及品牌行銷等提昇產品價值的工作，仍需要利用科技輔助來推進。

在這其中最需要改善的是芒果採收後的處理技術。愛文芒果採收後依品質高低篩選為 A、B、C 三等級，依序為出口用、內銷用、加工用。然而，愛文芒果依靠人工篩選，除了農村人口流失導致人力短缺，篩果流程也會因為保鮮期壓縮地極短，導致篩果階段約有 10% 左右的誤差，若以外銷金額估計，每年恐怕損失 1600 萬台幣。

2. Method

2.1 Task Definition

本資料集來源於日前組員參加之競賽，為一項科技部與農委會之合作計畫。因為農村人口流失導致人力短缺且篩果流程因保鮮期壓縮地極短，我們希望透過深度學習建立影像辨識演算法模型來對對愛文芒果的影像進行三種等級的自動分類。

2.2 Approach

2.2.1 Data Pre-processing

在一開始，我們將圖片大小壓縮至 800*800 訓練，但是在實際進行訓練後發現 800*800 的照片似乎有些過大，僅僅使用 300 張照片進行訓練便會將 12 GB 的 RAM 塞滿，於是我們決定進行圖片壓縮。在使用 100*100 的大小後，我們發現圖片會損失過多訊息，如下圖1；在參考了網路上其他 pretrained model 後，發現最常見的 224*224 效果最好，因此我們決定將圖片大小壓縮為最適中的 224*224，如下圖2。

圖1 100*100 芒果示意圖

圖2 224*224 芒果示意圖

考量到圖片背景可能會成為影響結果之雜訊，為了使圖片的重點更加聚焦於芒果本身，我們使用了物件偵測的技術，方法為 YOLOv5（You Only Look Once）。

為了實作 YOLOv5 模型以利篩選出每張圖之主題芒果，我們透過 labelImg 軟體與線上之芒果資料集共 1500 張圖片作為 YOLOv5 模型之訓練資料集；透過得到的權重檔套用至 YOLOv5 之 detect.py 中進行芒果偵測。

考量偵測物體通常位於照片正中央的概念，我們透過捕捉涵蓋圖片中心之物體，並記錄其座標進行圖片裁剪，如此便能得到聚焦的芒果圖片。

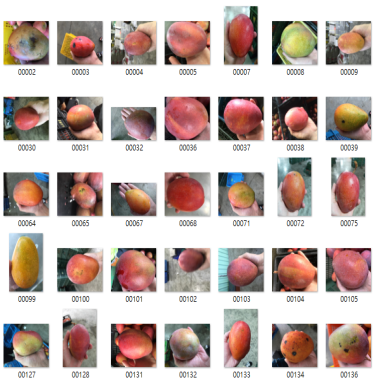

圖3 未經 YOLO 模型處理之圖片

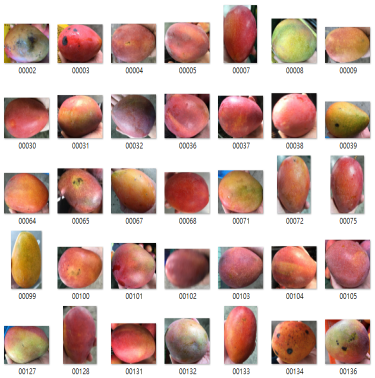

圖4 經 YOLO 模型處理之圖片

2.2.2 Model

首先，由於本次實驗的目的為透過圖片資料進行等級分類，我們可以將其視為一個的影像識別分類任務。基於其資料性質，我們決定採用卷積神經網路(Convolutional Neural Networks)，在後續的敘述都會將其簡稱為 CNN。接下來，我們會分別介紹 CNN 模型的搭建與和參數的調整方法，並套用已知的預訓練模型並進行微調。

在基本的 CNN 上，我們選擇實作對 Python 的支援性高的 Pytorch package，在 Google colab 上使用 GPU 進行訓練。在 batch size 為 100 的情況下，一共訓練 100 個 epoch 並將 early stop 設為 10。損失函數採用 categorical cross entropy，Optimizer 為 Adam、learning rate 為 0.001、dropout 為 0.25 且 weight decay 為 0.001。我們一開始使用作業練習過的 CNN 模型作為基礎，起初精準度約為 71% 左右；我們嘗試各種改進方法：如添加、減少層數與神經元等等。

如圖5，此 CNN 模型一共有四層卷積層和三層池化層，前者範圍大小為 3*3，而後者為 2*2，在每一次卷積後都會採用 ReLu 激活函數。神經元數量每層分別為 32、32、32 和 64，四層全連接層的神經元數量分別為 1024、256、128、64 和 3，最後一層的激活函數為 softmax，輸出三種類別的機率。

此模型在測試資料集得到了約 75.13% 的準確度，但我們認為模型仍提升的空間。因此，我們決定採用經過 YOLOv5 模型裁切後的資料再次進行 CNN 模型的訓練。

如圖6，在使用經過物件偵測(YOLOv5)裁減的照片後，我們的 CNN 在訓練過程中可以明顯看到損失函數下降地更快並取得不錯的結果，精準度提升至大約 76.13%(提升 1%)。然而，我們仍然希望可以取得更加滿意的結果，因此我們決定使用已知的預訓練模型以提高準確率。

我們使用預訓練的 VGG-16 模型進行訓練，在模型架構上，透過 torchvision 套件所提供的模型參數，並接上三層的全連階層網路配合 dropout 和 batch normalization 來預測出芒果的三種類別，模型整體架構如圖7所示。在訓練階段部分，將圖片大小轉換成適合的長寬，同時進行正規化使得模型訓練更加穩定，減少圖片亮暗與色彩所造成之影響，並將圖片放入模型中進行訓練，而模型精確度達到了 79.37%，優於 CNN 模型 3%，顯示使用預訓練模型可以達到更好的預測結果。

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 222, 222]	896
BatchNorm2d-2	[-1, 32, 222, 222]	64
ReLU-3	[-1, 32, 222, 222]	0
Conv2d-4	[-1, 32, 220, 220]	9,248
BatchNorm2d-5	[-1, 32, 220, 220]	64
ReLU-6	[-1, 32, 220, 220]	0
MaxPool2d-7	[-1, 32, 110, 110]	0
Conv2d-8	[-1, 32, 108, 108]	9,248
BatchNorm2d-9	[-1, 32, 108, 108]	64
ReLU-10	[-1, 32, 108, 108]	0
MaxPool2d-11	[-1, 32, 54, 54]	0
Conv2d-12	[-1, 64, 52, 52]	18,496
BatchNorm2d-13	[-1, 64, 52, 52]	128
ReLU-14	[-1, 64, 52, 52]	0
MaxPool2d-15	[-1, 64, 26, 26]	0
Linear-16	[-1, 1024]	44,303,360
Dropout-17	[-1, 1024]	0
ReLU-18	[-1, 1024]	0
Linear-19	[-1, 256]	262,400
Dropout-20	[-1, 256]	0
ReLU-21	[-1, 256]	0
Linear-22	[-1, 128]	32,896
Dropout-23	[-1, 128]	0
ReLU-24	[-1, 128]	0
Linear-25	[-1, 64]	8,256
Dropout-26	[-1, 64]	0
ReLU-27	[-1, 64]	0
Linear-28	[-1, 3]	195

圖5 CNN 模型架構

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 222, 222]	896
BatchNorm2d-2	[-1, 32, 222, 222]	64
ReLU-3	[-1, 32, 222, 222]	0
Conv2d-4	[-1, 32, 220, 220]	9,248
BatchNorm2d-5	[-1, 32, 220, 220]	64
ReLU-6	[-1, 32, 220, 220]	0
MaxPool2d-7	[-1, 32, 110, 110]	0
Conv2d-8	[-1, 64, 108, 108]	18,496
BatchNorm2d-9	[-1, 64, 108, 108]	128
ReLU-10	[-1, 64, 108, 108]	0
MaxPool2d-11	[-1, 64, 54, 54]	0
Conv2d-12	[-1, 64, 52, 52]	36,928
BatchNorm2d-13	[-1, 64, 52, 52]	128
ReLU-14	[-1, 64, 52, 52]	0
MaxPool2d-15	[-1, 64, 26, 26]	0
Linear-16	[-1, 1024]	44,303,360
Dropout-17	[-1, 1024]	0
ReLU-18	[-1, 1024]	0
Linear-19	[-1, 256]	262,400
Dropout-20	[-1, 256]	0
ReLU-21	[-1, 256]	0
Linear-22	[-1, 128]	32,896
Dropout-23	[-1, 128]	0
ReLU-24	[-1, 128]	0
Linear-25	[-1, 64]	8,256
Dropout-26	[-1, 64]	0
ReLU-27	[-1, 64]	0
Linear-28	[-1, 3]	195

Total params: 44,673,059
Trainable params: 44,673,059
Non-trainable params: 0

Input size (MB): 0.57
Forward/backward pass size (MB): 97.33
Params size (MB): 170.41
Estimated Total Size (MB): 268.32

圖6 CNN + YOLO 模型架構

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 224, 224]	1,792
ReLU-2	[-1, 64, 224, 224]	0
Conv2d-3	[-1, 64, 224, 224]	36,928
ReLU-4	[-1, 64, 224, 224]	0
MaxPool2d-5	[-1, 64, 112, 112]	0
Conv2d-6	[-1, 128, 112, 112]	73,856
ReLU-7	[-1, 128, 112, 112]	0
Conv2d-8	[-1, 128, 112, 112]	147,584
ReLU-9	[-1, 128, 112, 112]	0
MaxPool2d-10	[-1, 128, 56, 56]	0
Conv2d-11	[-1, 256, 56, 56]	295,168
ReLU-12	[-1, 256, 56, 56]	0
Conv2d-13	[-1, 256, 56, 56]	590,080
ReLU-14	[-1, 256, 56, 56]	0
Conv2d-15	[-1, 256, 56, 56]	590,080
ReLU-16	[-1, 256, 56, 56]	0
MaxPool2d-17	[-1, 256, 28, 28]	0
Conv2d-18	[-1, 512, 28, 28]	1,180,160
ReLU-19	[-1, 512, 28, 28]	0
Conv2d-20	[-1, 512, 28, 28]	2,359,808
ReLU-21	[-1, 512, 28, 28]	0
Conv2d-22	[-1, 512, 28, 28]	2,359,808
ReLU-23	[-1, 512, 28, 28]	0
MaxPool2d-24	[-1, 512, 14, 14]	0
Conv2d-25	[-1, 512, 14, 14]	2,359,808
ReLU-26	[-1, 512, 14, 14]	0
Conv2d-27	[-1, 512, 14, 14]	2,359,808
ReLU-28	[-1, 512, 14, 14]	0
Conv2d-29	[-1, 512, 14, 14]	2,359,808
ReLU-30	[-1, 512, 14, 14]	0
MaxPool2d-31	[-1, 512, 7, 7]	0
AdaptiveAvgPool2d-32	[-1, 512, 7, 7]	0
Linear-33	[-1, 512]	12,845,568
Dropout-34	[-1, 512]	0
Linear-35	[-1, 256]	131,328
Dropout-36	[-1, 256]	0
BatchNorm1d-37	[-1, 256]	512
Linear-38	[-1, 3]	771

Total params: 27,692,867
Trainable params: 27,692,867
Non-trainable params: 0

圖7 VGG-16 模型架構

3. Data collection and validation

圖片拍攝自合作果園，本資料集為盡量模仿真實應用場域，考量到芒果處理廠的人力、水果保鮮時間、現地設備等諸多限制，以中低成本之一般消費型相機／攝影機蒐集大量圖片。

此外，考量利用相機一張一張拍攝獲取資料的方式效率低落，顯然不足以滿足未來自動智慧篩果機器所需，我們部分資料便利用攝影機錄影擷取芒果影像的方式，來模擬真實應用場域情境。藉由拍攝影片，我們預期蒐集的芒果照片將會有部分移動模糊（Motion Blur）、成像雜訊（Sensor Noise）以及光照變化（Luminance Variant）等情況。

兩份資料集均包含了完整的影像圖檔和各影像對應之等級標籤，呈現於下方圖8與圖9。從途中可以得知，不管是在訓練資料集還是在測試資料集上，此資料集的資料分布都相當平均，不存在資料不平衡的問題。在實作上，我們會將測試資料集的 5600 筆資料以 4 比 1 切成測試資料集和驗證資料集，分別為 4480 和 1120 筆進行訓練。

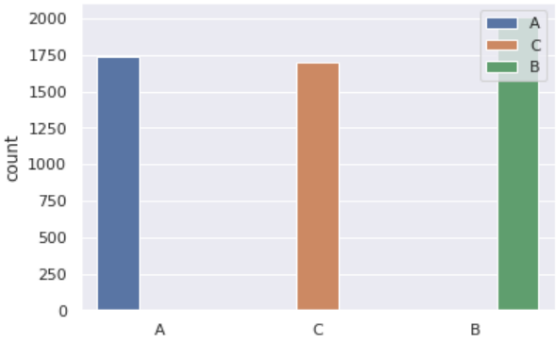

圖8 Training Set 5600 筆

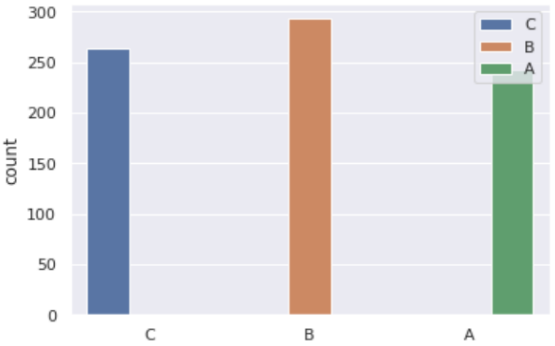

圖9 Testing Set 800 筆

4. Evaluation

4.1 Metrics

在評論模型效能的指標部分，我們主要會使用兩種指標，包含 accuracy 和 F1-score。其中，F1-score 指標可以想像成 precision 和 recall 的調和

平均。我們紀錄的 F1-score 包含了每個標籤個別的和總體的分數，目的是從中觀察每個類別分類表現的差異。

4.2 Baselines

在本次實驗中，我們將基本的 CNN 模型視為 baseline。在提升效能的方面上，我們實作了擷取圖片重點的 YOLOv5 和對預訓練模型 VGG16 進行 fine-tuning，希望能夠在分類任務上取得更好的成果。

4.3 Results

本次的實驗結果紀錄於表1。首先，我們的基本模型在經過複雜的調整以後，其實已經足以處理本資料集的資料了，因此，在經過套用 YOLOv5 演算法後取得的提升並不顯著。儘管如此，我們仍然認為類似 YOLOv5 這種物件偵測的技術事十分實用的，透過幫助模型專注於物件本身的圖片、不會被背景或其他物件干擾便可以免除許多不必要的資訊。相比起經過不斷參數調整才有良好成效的基礎模型，單單只是套用這種演算法便能使模型提升效能就已經證明其泛用性和實用性。

儘管 VGG16 與 CNN 相比模型複雜許多，且使用的也是經過 pre-train 的模型，但表現也只比 CNN 好上些許，我們認為這是因為芒果分級不需過於複雜的模型。此外，VGG16 pre-train 的資料集主要為日常生活看到的事物，且訓練目的為分辨不同種類的物件，與芒果分級較無關係。

我們認為其中最值得探討的一點是，三種不同等級芒果的 F1-score 相差許多，其中 C 表現的最佳，A 次之，B 則最差。我們認為會造成此現象的有兩個原因。首先，若是芒果本身賣像不佳，很容易就可以將其分類至 C 等級。因此，模型學習圖片上黑點或瑕疵的區塊剛好吻合人工識別芒果等級的方式。再來，B 等級身為 A 和 C 的過渡等級，在人工識別時就極有可能因為不同的標準造成 B 等級本身的品質並不統一，使模型在學習時難以有一個衡量的標準，造成此等級的分類效果不佳。

表 1 實驗結果

	CNN	CNN + YOLO	VGG16	VGG16 + YOLO
Acc.	0.7513	0.7613	0.7937	0.7637
F1-A	0.7542	0.7724	0.7671	0.7985
F1-B	0.6822	0.6861	0.7425	0.6759
F1-C	0.8235	0.8399	0.8790	0.8291
F1	0.7533	0.7661	0.7962	0.7678

5. Conclusion

由於每個模型分類器之間必定有些差異，若是可以在這些模型中截長補短，多訓練一點模型做 ensemble learning，也許能就能夠學習到錯誤分類資料的特性，進而提升分類結果。

另外，也許嘗試其他更加新穎的網路架構，或許能夠解決深度學習神經網路的退化問題，取得更好的效果。

6. 分工表格

	分工項目
黃柏叡	YOLOv5模型訓練、書面資料、投影片編輯、報告負責人
黃啟宏	資料集蒐集、CNN模型訓練、書面資料、投影片編輯
潘躍升	VGG16模型訓練、書面資料、投影片編輯