**農地作物現況調查影像辨識競賽**

報告說明文件

團隊編號:

Team\_1927

作者

賴冠羽 輔仁大學 數學系

王譽錚 輔仁大學 數學系

胡佩欣 輔仁大學 數學系

龔意晴 輔仁大學 數學系

何昱融 輔仁大學 數學系

指導老師:

陳泓勳老師

**目錄**

[壹、 環境](#_heading=h.gjdgxs) 2

[貳、 演算方法與模型架構](#_heading=h.30j0zll) 3

[參、](#_heading=h.1fob9te) 資料處理 [4](#_heading=h.1fob9te)

[肆、](#_heading=h.3znysh7) 訓練方式 [6](#_heading=h.3znysh7)

[伍、](#_heading=h.2et92p0) 分析與結論 [7](#_heading=h.2et92p0)

[陸、](#_heading=h.3dy6vkm) 程式碼 [9](#_heading=h.3dy6vkm)

[柒、](#_heading=h.1t3h5sf) 使用的外部資源與參考文獻 [13](#_heading=h.1t3h5sf)

[捌、](#_heading=h.4d34og8) 作者聯絡資料表 [13](#_heading=h.4d34og8)

環境

1. 作業系統：Windows10、Windows11
2. 顯示卡 : RTX 3090、GTX 1080、GTX 1080 Ti
3. VRAM :at least 8 GB
4. 語言：Python3.8
5. 套件：
6. numpy==1.22.3
7. pandas==1.1.3
8. os==3.8
9. matplotlib==3.3.2
10. shutil==3.8
11. opencv-python==4.6.0.66
12. scikit-learn==1.1.2
13. pillow==8.0.1
14. tensorflow==2.8.0
15. seaborn==0.11.0
16. glob==3.8
17. 預訓練模型
18. Xception
19. DenseNet121
20. DenseNet201

**演算方法與模型架構**

**資料擴增:**為了提升模型最後預測的一般性，本團隊在模型訓練前使用了 tensorflow.keras.preprocessing.image底下的ImageDataGenerator來增加訓練資料的多樣性。其中的參數為rotation\_range=20, horizontal\_flip=True, vertical\_flip=True, zoom\_range=0.2, rescale=1/255(若遷移式學習模型的前處理沒有做特徵壓縮的話才加入此參數),preprocessing\_function=preprocess\_input(preprocess\_input為遷移式學習模型的對應前處理)

**學習率調整:**本團隊使用了tensorflow.keras.callbacks.LearningRateScheduler()函數來隨著訓練輪數的增加降低學習率以達到準確率的最佳值。其中初始學習率為0.001，當訓練輪數超過13輪時調整為0.0005;當訓練輪數超過17輪時調整為0.0001;最後超過25輪的時候調整為0.00001。

**保存模型:**為確保最後完成訓練的模型有被儲存，本團隊利用tensorflow.keras底下的callbacks.ModelCheckpoint()方法在訓練的過程中就依據val\_accuracy的最佳值保存模型。

**編譯模型:**在編譯模型的時候本團隊利用了categorical\_crossentropy做為訓練過程中的損失函數、優化器使用了tenforflow.keras.optimizers底下的Adam優化器。

**批次訓練尺寸(Batch\_size):**由於設備的VRAM空間有限，所以本團隊根據資料前處理後不同的資料尺寸來決定batch size。若經過前處理的資料尺寸為400\*400 pixels的話，batch size設為32或16;若為資料尺寸600\*600 pixels的話，batch size設為16或8;若為資料尺寸800\*800 pixels的話，batch size設為8。

**訓練輪數(Epochs):**為了避免模型過擬合(overfitting)以及訓練時間過長的情況發生，本團隊在這次競賽所訓練的輪數都不超過30輪以確保過擬合的情況發生，同時也可以在一定的時間內得到訓練結果。

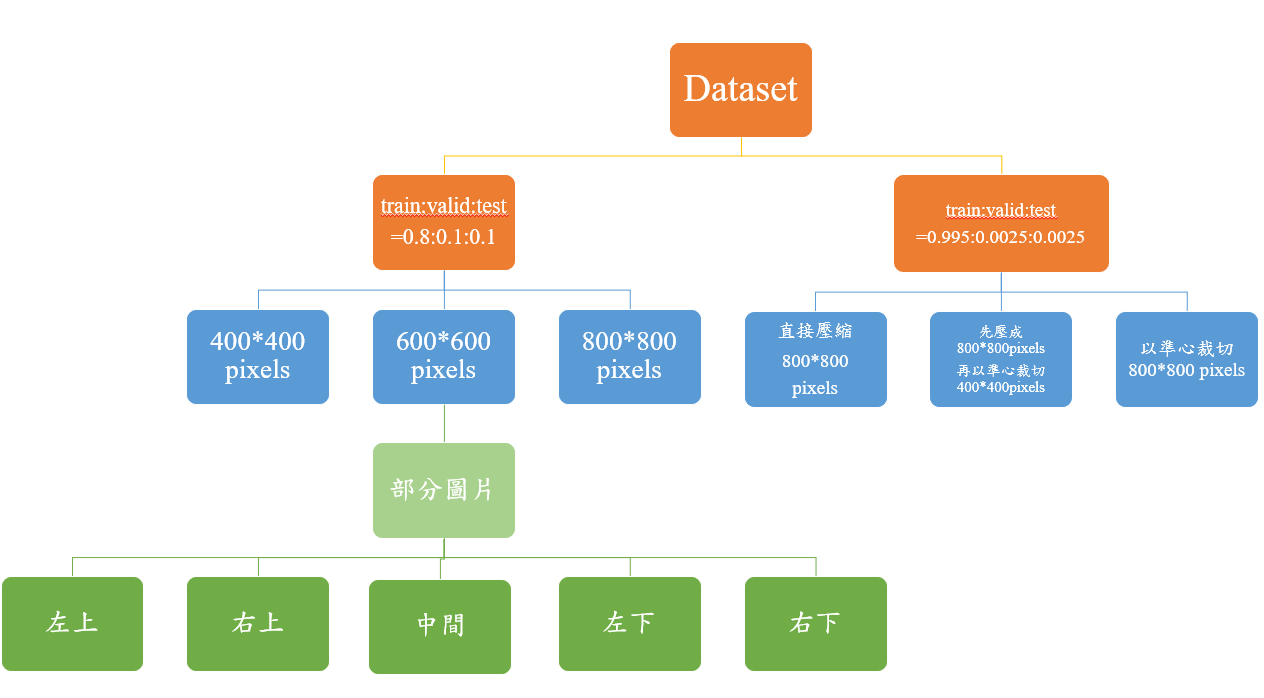
**模型架構:**本團隊在此次競賽主要是採取遷移式學習的方式來訓練模型。根據不同的前處理模式使用了不同的預訓練模型，而最終版本使用了Xception, DenseNet121, DenseNet201。模型資訊如下表並參照了Keras Application。

| 模型名稱 | 大小(MB) | 參數量 | 模型深度 | 運算時間(ms/step) |
| --- | --- | --- | --- | --- |
| Xception | 88 | 22.9M | 81 | 8.1 |
| DenseNet121 | 33 | 8.1M | 242 | 5.4 |
| DenseNet201 | 80 | 20.2M | 402 | 6.7 |

**資料處理**

在本次競賽中，我們對於資料集的處理主要分為訓練前的資料前處理，以及訓練後的資料合併兩大部分。

在資料前處理的部分，我們對圖片進行切割與壓縮。處理過程中發現官方提供的資料集中，圖片大小與準心位置差異太大，因此我們將圖片裁切縮放至合適的尺寸與準心位置以達到更好的訓練效果。而切割的方式則如下圖並說明之:

**Figure 1** : 資料處理流程圖

首先，在裁切壓縮過程中，為了提高準確率，我們分成了兩次試驗：

1. 對於Dataset、Public 和Private原圖檔進行三種裁切尺寸：

第1種：依照準心裁切成尺寸為400\*400 pixels的圖檔

第2種：依照準心裁切成尺寸為600\*600 pixels的圖檔

(額外依照準心分別向左上、左下、右上、右下裁

切成尺寸為600\*600 pixels的圖檔)

第3種：依照準心裁切成尺寸為800\*800 pixels的圖檔

1. 對於Dataset、Public和Private原圖檔進行三種處理方式：

第1種：依照準心裁切成尺寸為800\*800 pixels的圖檔

第2種：直接壓縮成800\*800 pixels的圖檔

第3種：壓縮成800\*800 pixels的圖檔，再依照準心裁切

400\*400 pixels的圖檔

接著在資料合併的部分，我們將三種模型預測結果進行合併，並根據投票的方式，選擇投票機率最高的類別作為圖片最終預測答案，以提高準確率。

在競賽初期，我們將Dataset切分成train、valid、test，圖片張數比例為0.8:0.1:0.1的Trainset，並使用400\*400 pixels、600\*600 pixels、800\*800 pixels的圖檔訓練模型。為了提高準確率，競賽後期我們將Dataset切分成圖片張數比例為0.995:0.0025:0.0025的Trainset，並分別對三種處理方式的Trainset進行模型訓練；同時也將Public和Private原圖檔根據Trainset所處理的方式進行裁切縮放，並使用Test Time Augmentation(數據增強，簡稱TTA)方法預測結果，進行資料合併，選擇機率最高的類別作為圖片最終預測答案。

接著我們利用DataFrame讀取官方提供的csv檔，抓出其中的地理位置資訊，統計Dataset每個縣市出現的總次數(第3列)，並交叉比對每個縣市出現前五多的種類，將縣市總出現的次數扣掉前五多種類的張數(第0列)，進而統計出各縣市不在前五多的種類占此縣市機率(第4列)。由於新北市、台北市和宜蘭縣出現總類少於五種，因此機率為零就不列在下方的圖片。



**訓練方式**

本次競賽我們採用的訓練方法為遷移式學習，在訓練過程中我們使用不同的資料處理方式並分別使用預訓練模型Xception, DenseNet201, DenseNet121訓練資料，同時過程中我們使用凍結訓練。訓練的過程和方法如下 :

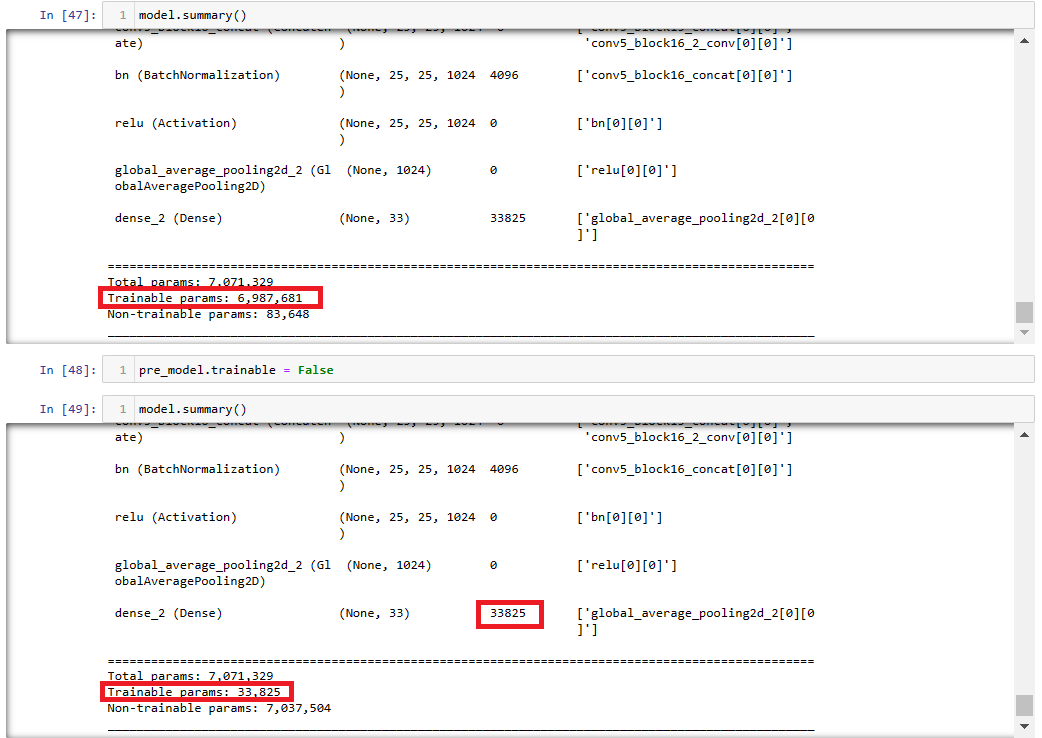
1. : 設定模型及調整參數

* 首先我們將以下三種資料處理方式分別放入不同預訓練模型中，並依圖片裁切尺寸設定不同batch size。若圖片尺寸為800\*800 pixels，則batch size設定為8；若圖片尺寸為400\*400 pixels，則batch size設定為16。

|  | 資料處理方式 | 對應預訓練模型 | 簡稱 | batch size |
| --- | --- | --- | --- | --- |
| 方法一 | 依照準心裁切成尺寸為800\*800 pixels | Xception | crop模型 | 8 |
| 方法二 | 直接壓縮成800\*800 pixels | DenseNet201 | org模型 | 8 |
| 方法三 | 壓縮成800\*800 pixels，  再依準心切400\*400 pixels | DenseNet121 | org\_crop模型 | 16 |

1. : 凍結訓練

接下來我們將Epoch設定訓練20輪，並採用預訓練模型的結構，將預訓練模型凍結(pre\_model.trainable=False)，以DenseNet121為例，凍結後模型只有最後一層被訓練，訓練參數減為最後一層數量。同時將初始學習率設定為為0.001，當訓練輪數超過13輪時調整為0.0005、超過17輪時調整為0.0001。

1. ：模型再凍結

* Epoch=20輪訓練完後，我們將訓練模型解開凍結，並排除預訓練模型末五層再次進行凍結，將末五層參數保留再進行10輪訓練。同時當訓練輪數超過25輪時，學習率則調整為0.00001。

1. ：觀測結果

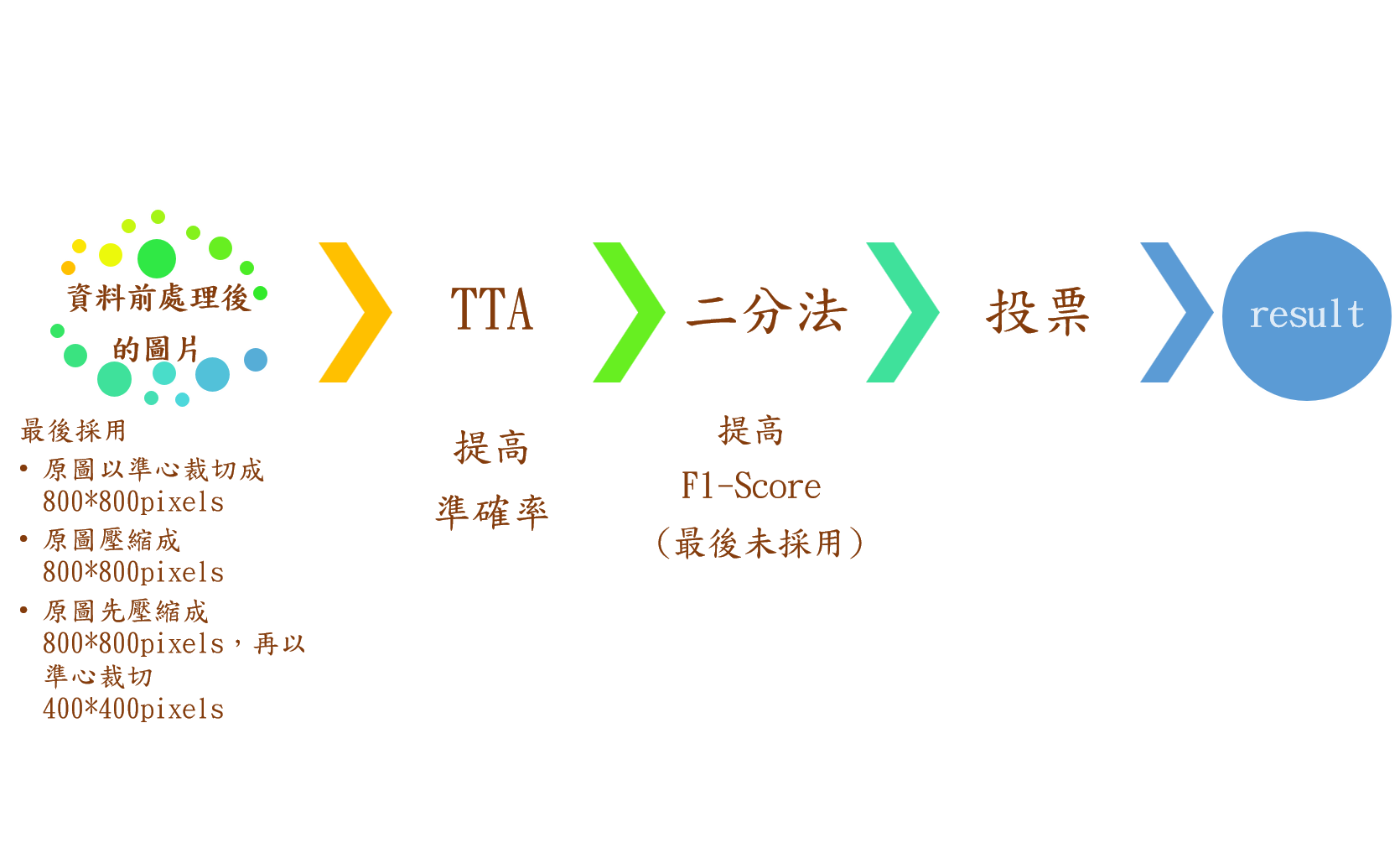
* 模型訓練完後，我們以test資料集進行預測，使用sklearn.metrics套件繪製混淆矩陣及計算準確率，並顯示分類指標精確率、召回率及F1-score以了解模型訓練是否達到理想。

1. ：儲存模型

* 為確保最後完成訓練並達到理想的模型有被儲存，最後我們依據val\_accuracy的最佳值保存模型。

**分析與結論**

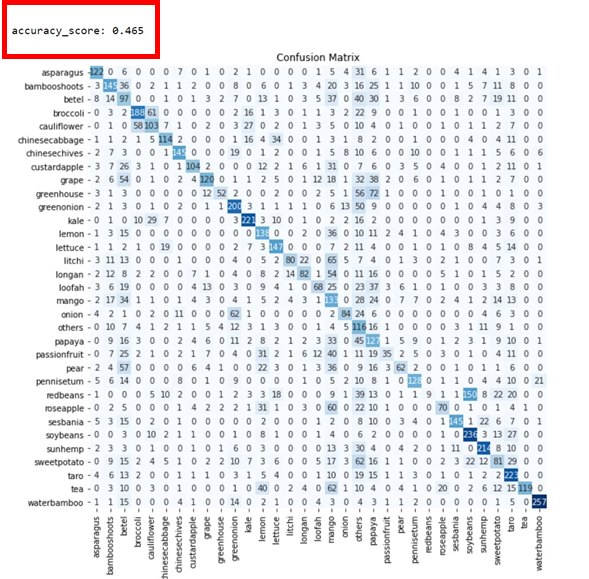
在本次競賽過程中，為了解決我們遇到的問題，例如：提高準確率，提高F1-score，採用了許多不同的方法訓練模型，也根據不同方法的結果決定是否採用該方法，而訓練過程如下:



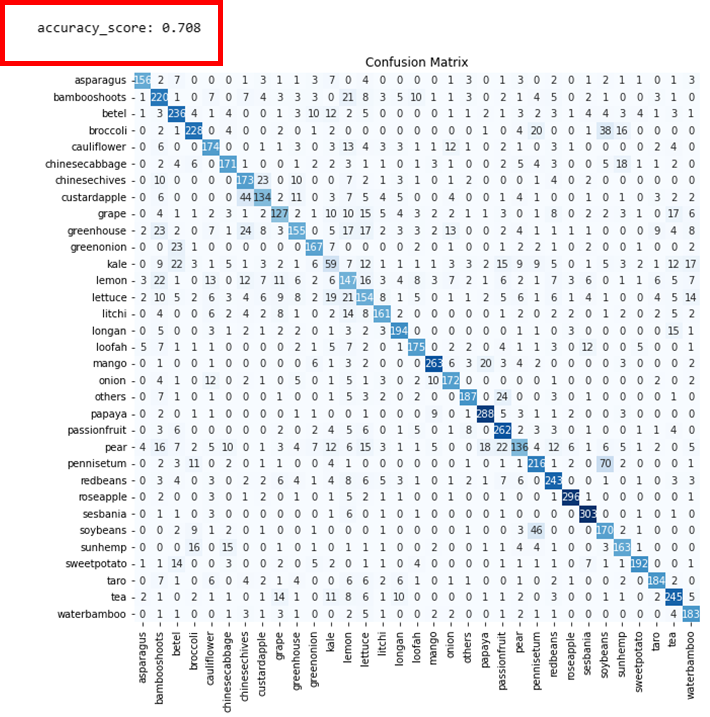
**Figure 2** : 最終流程

各方法的使用的原因及分析如下：

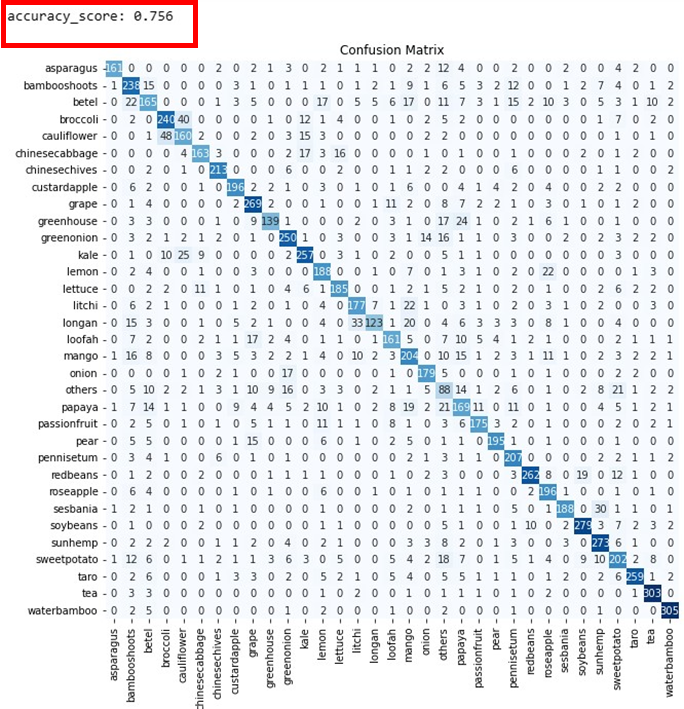
1. 我們先依照準心將原圖裁切為400\*400 pixels(簡稱裁切400模型)、600\*600 pixels(簡稱裁切600模型)、800\*800 pixels(簡稱裁切800模型)三種尺寸的Dataset，再將個別Dataset資料切割成train:test:valid=0.8:0.1:0.1，放入Xception預訓練模型進行訓練。



**Figure3** ：裁切400模型之混淆矩陣，準確率為0.465



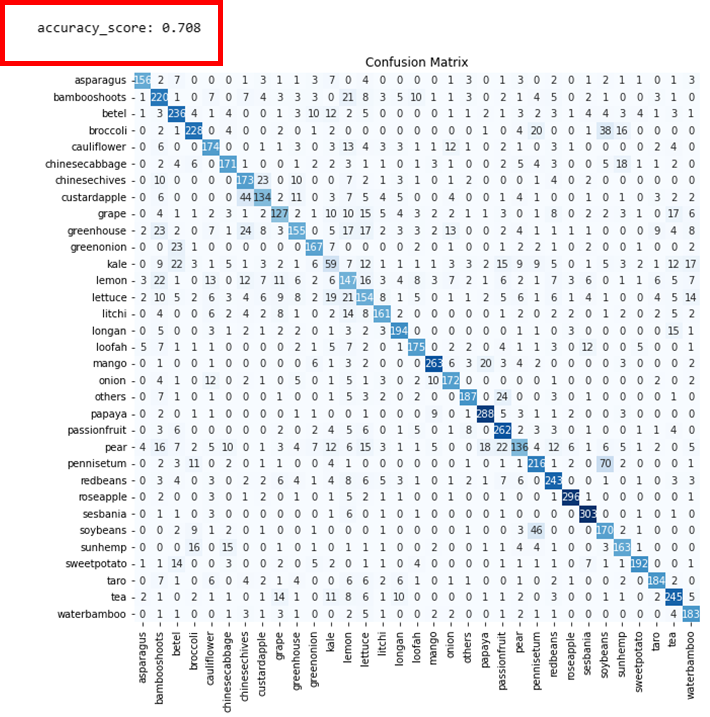
**Figure 4**：裁切600模型之混淆矩陣，準確率為0.708



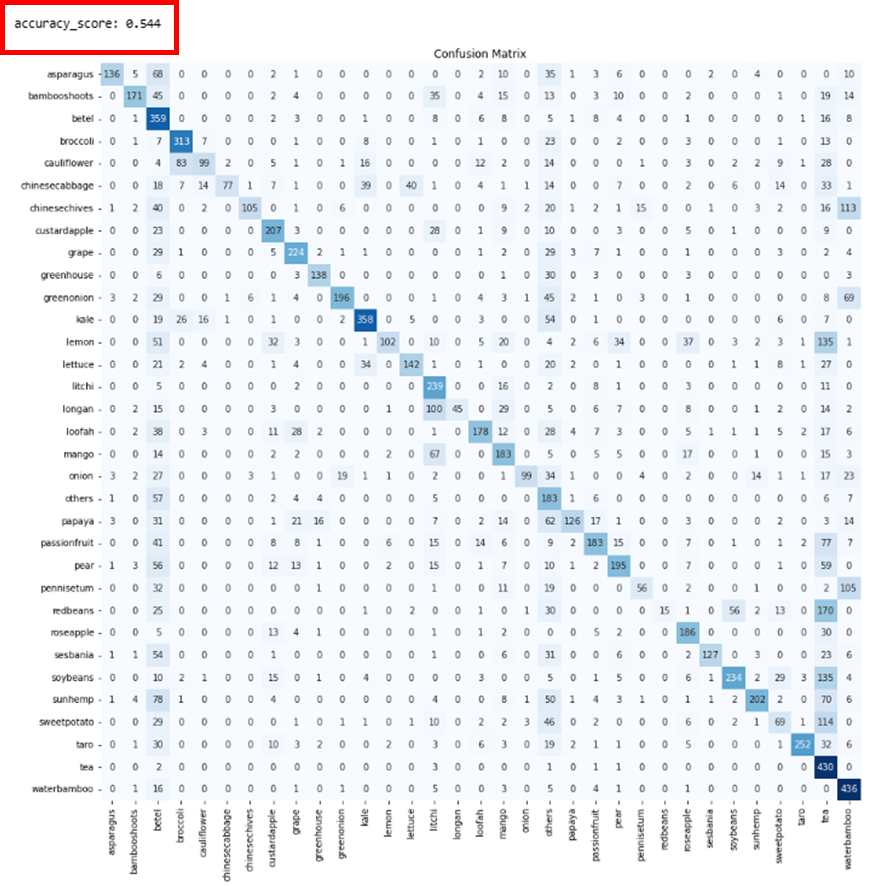
**Figure 5**：裁切800模型之混淆矩陣，準確率為0.756

由Figure 3至Figure 5得知，三種尺寸的模型準確率分別為0.465、0.708、0.756，其中裁切800模型的準確率最高，所以最後僅採用800\*800 pixels的尺寸。

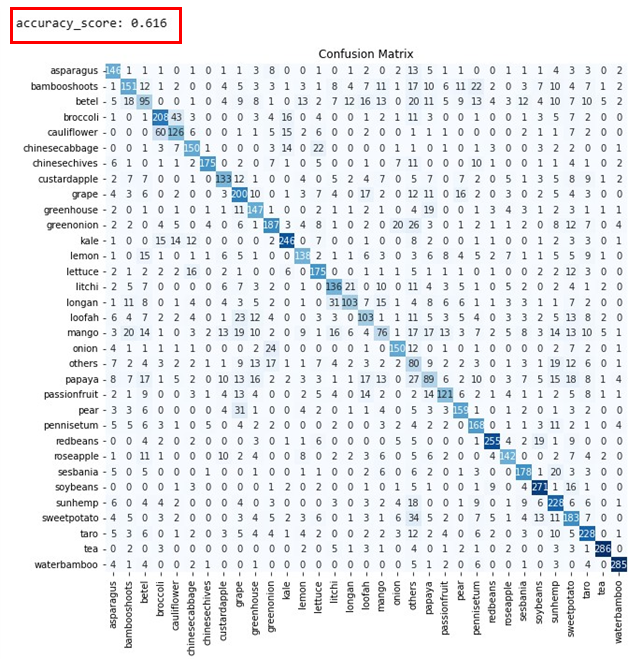
2. 其中針對600\*600 pixels的圖檔局部切割為：左上(600\_lu模型)、左下(600\_ld模型)、右上(600\_ru模型)、右下(600\_rd模型)、中間(600\_模型)五種局部圖片，分別進行模型訓練，想藉此瞭解圖片的局部特徵是否能被預測為正確類別。



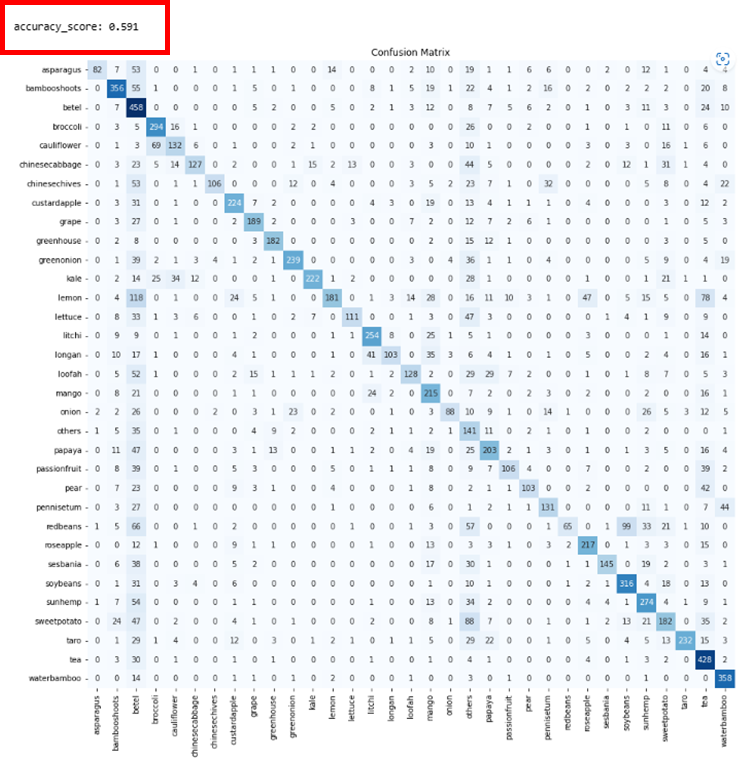
**Figure 6**：600\_模型之混淆矩陣，準確率為0.708



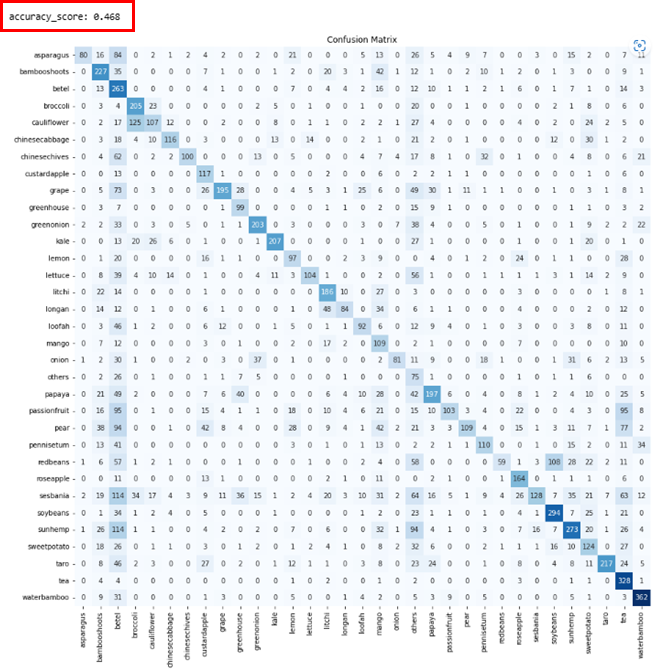
**Figure 7**：600\_lu模型之混淆矩陣，準確率為0.544



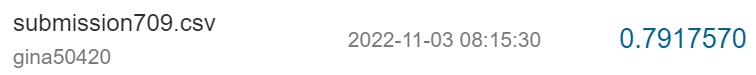
**Figure 8**：600\_ld模型之混淆矩陣，準確率為0.616



**Figure 9**：600\_ru模型之混淆矩陣，準確率為0.591



**Figure 10**：600\_rd模型之混淆矩陣，準確率為0.468



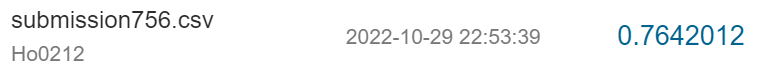
**Figure 11**：600\_模型預測公開集上傳成績



**Figure 12**：五種位置的模型預測公開集合併後上傳成績

從Figure 6至Figure 10得知，除了中間位置(600\_模型)的準確率較高，其餘四種位置的準確率皆不到0.7。也因此我們分別將五種模型預測公開集，隨後選擇出現較多的預測值作為答案。由Figure 11與Figure 12可知，雖然使用五種位置預測的方法有提高成績，但成效不顯著，所以決定不採用這個方法。

3. 在預測資料的時候，我們利用了TTA的方法來增加預測的準確率。這個方法可以將同一張圖片透過不同的轉換方式讓模型利用不同的資訊擴增來預測同一張圖片，最後再取出模型判斷最多相同答案的作為這一張圖片的最終答案。我們在這個方法中讓模型判斷原圖、水平和垂直翻轉後的圖、旋轉90度和270度的圖、以及增強對比過後的圖，利用以上6種方法來讓模型預測同一張圖片。



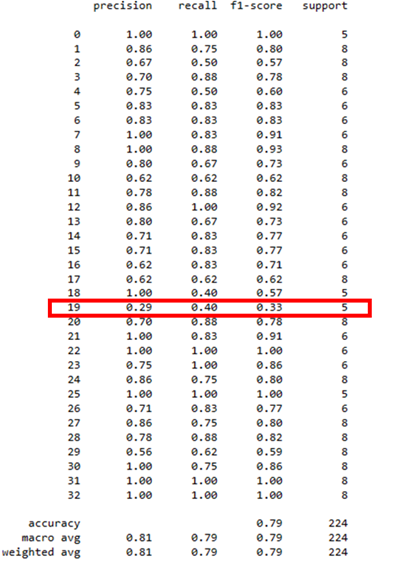
**Figure 13**：裁切800模型預測公開集上傳成績



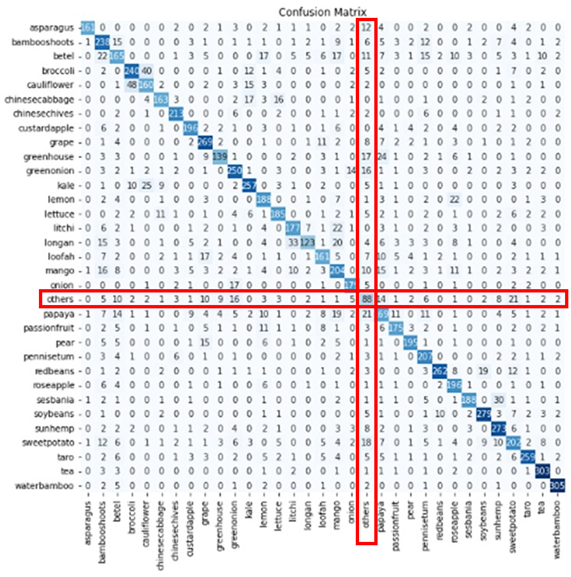
**Figure 14**：裁切800模型使用TTA預測公開集上傳成績

對比Figure 13與Figure 14，使用TTA方法可有效提高預測的準確率，因此在之後的實驗皆有使用這個方法。

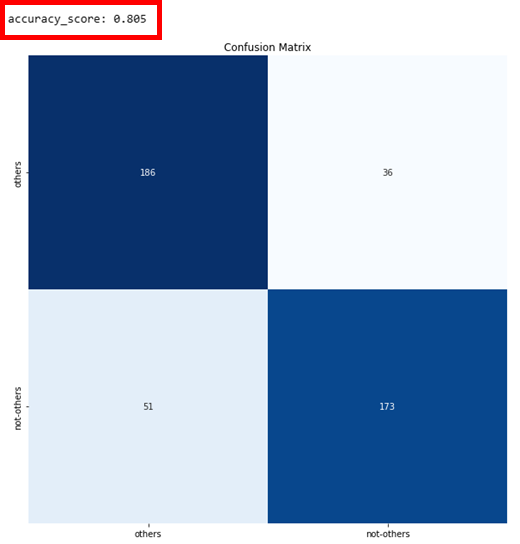
4. 以裁切800模型為例，由Figure 15、Figure 16可得知others易被誤判成其他類別，準確率、F1-Score跟其他類別相較之下也較低，所以我們特別將others抓出來，其餘32個類別隨機挑選固定比例數量的圖檔作為非others類別，並訓練模型others與非others(簡稱others模型)，嘗試提高預測為others的F1-Score。

****

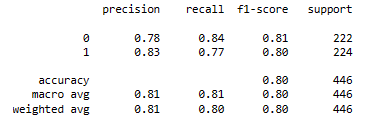
**Figure 15**：裁切800模型之各類別F1-Score



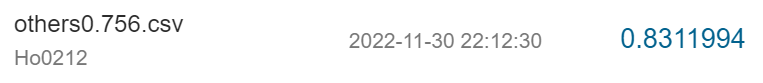
**Figure 16**：裁切800模型之混淆矩陣



**Figure 17**：others模型之混淆矩陣，準確率為0.805



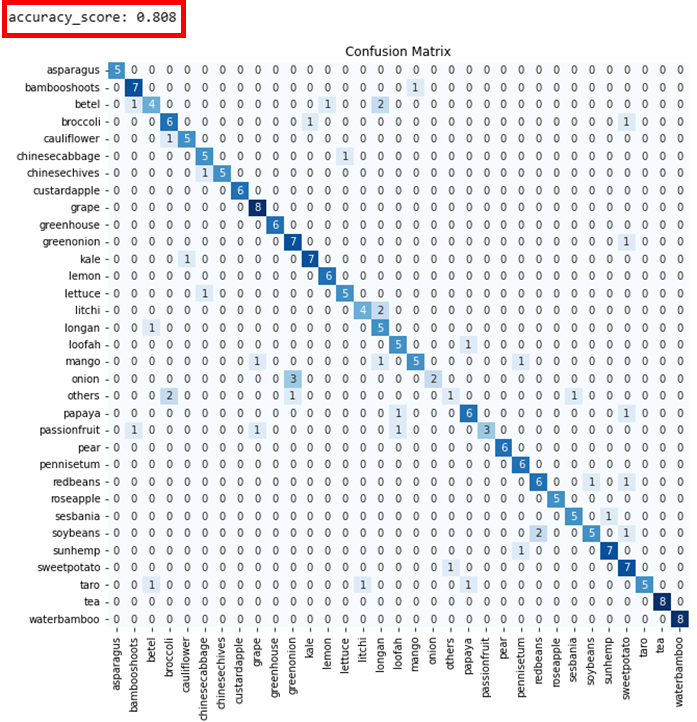
**Figure 18**：others模型之F1-Score



**Figure 19**：裁切800模型加others模型，使用TTA預測公開集上傳成績

從Figure 17可看出，others模型對於區分others與非others的方法效果顯著，且在Figure18中看出F1-score有顯著提升，但對比Figure 14與Figure 19，使用others模型的預測成績相比未使用反而降低一些，所以最後未採用此方法。

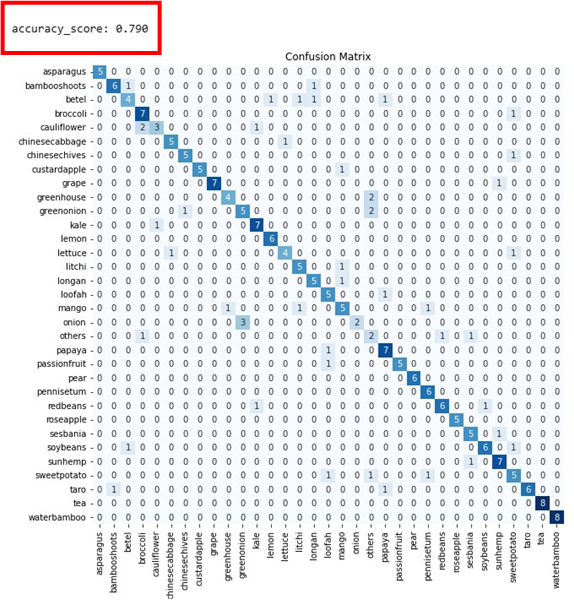
5. 將確定好尺寸的Dataset分成train:test:valid=0.995:0.0025:0.0025，圖檔進行資料擴增後，依續放入三種不同的預訓練模型進行訓練，分別為DenseNet201、DenseNet121、Xceptioon，準確率為0.808、0.79、0.732。

****

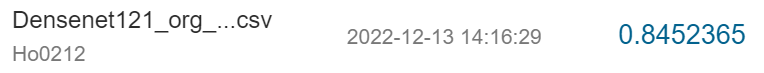
**Figure 20：**org模型之混淆矩陣 **(DenseNet201)**



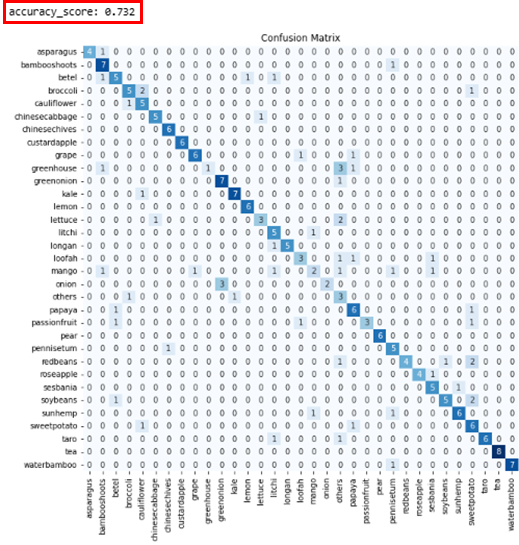
**Figure 21**：org模型，使用TTA預測公開集後上傳成績

****

**Figure 22：**org\_crop模型之混淆矩陣 **(DenseNet121)**



**Figure 23**：org\_crop模型，使用TTA預測公開集後上傳成績

****

**Figure 24：**crop模型之混淆矩陣 **(Xception)**



**Figure 25**：crop模型，使用TTA預測公開集上傳成績

6. 最終我們把公開集資料放入步驟5的三種模型中，並皆使用TTA方法進行預測，接著將三種結果進行投票。一張圖中有兩個以上模型預測結果相同，則採用相同的結果，若三個預測結果皆不同，則採用模型中成績最高的org模型的預測結果作為最終答案。

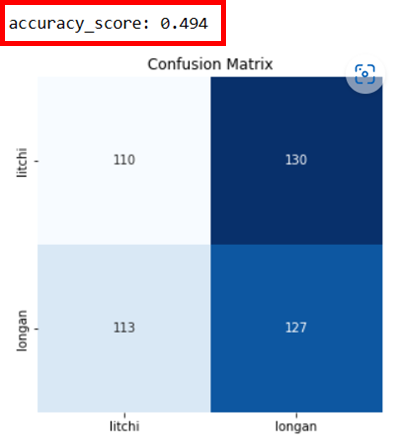


**Figure 26**：三種模型的公開集預測結果投票後上傳成績

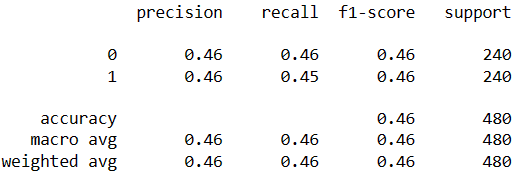
從Figure 26可知，TTA與投票二種方法皆能有效提高預測準確率。當圖片利用TTA進行6種方法之預測，能夠多次驗證預測結果的準確性，同時透過投票也能讓些許答案更加精準被預測。因此最終版本中，我們如法炮製地將三種不同模型進行TTA預測及投票，也使得最終預測成績有明顯提升。

**改進方向：**

1.二分法訓練:

在模型訓練的過程中，透過混淆矩陣，我們發現特定類別容易被互相混淆，例如：荔枝與龍眼、白花椰菜及綠花椰菜。因此我們使用多種二分法訓練易混淆的種類，讓預測public準確率更提升，並藉此提高誤判的F1-score。

**Figure 27**：荔枝與龍眼二分法訓練之混淆矩陣，準確率0.494

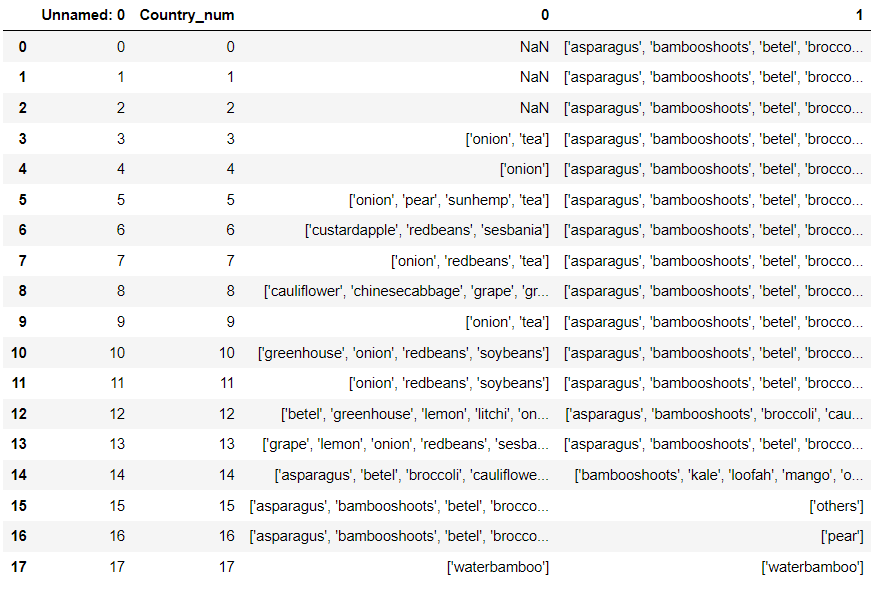
****

**Figure 28**：荔枝與龍眼二分法訓練F1-Score

透過Figure 27和Figure 28可以發現我們的訓練結果不盡理想。因此將二分類訓練進行更加改善的訓練，並利用前面針對模型改善的方法多加訓練，使準確率能夠再提升，以改進我們33多類別訓練過程的不足。

2.地理資訊:

原先我們預期能透過地理資訊刪除不可能在該縣市出現的種類，但根據官方所給的資料並分析後，有些縣市全部的農作物種類皆有可能出現(Figure 29)，且針對預測public成績最高的模型與官方給的Dataset答案進行分析結果卻不盡相同，例如:在宜蘭縣，Dataset中全部皆出現梨子，但在public資料中，宜蘭縣的圖經過我們的人工判斷為蔥。

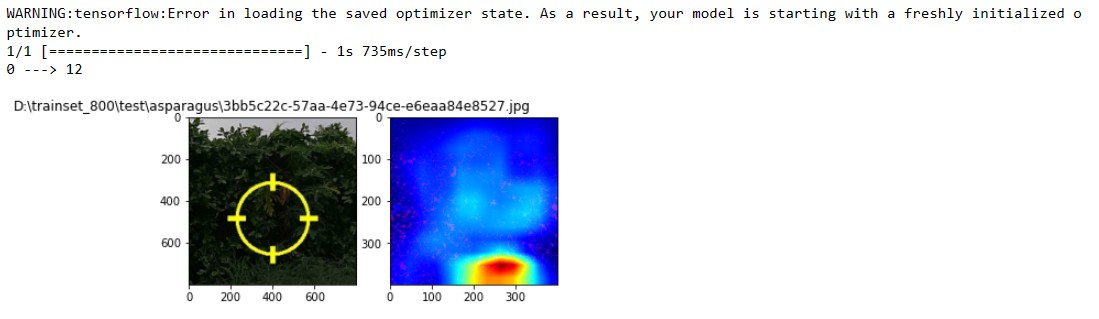


**Figure 29**：各縣市出現的農作物種類

因此藉由其他方式利用不同地理資訊過濾資料，例如:各地的盛產農作物(屏東盛產紅豆、台南盛產芒果......等)、不同緯度適合耕作的農作物不同來篩選，也是可以改進的方向。

3.Grad Cam:

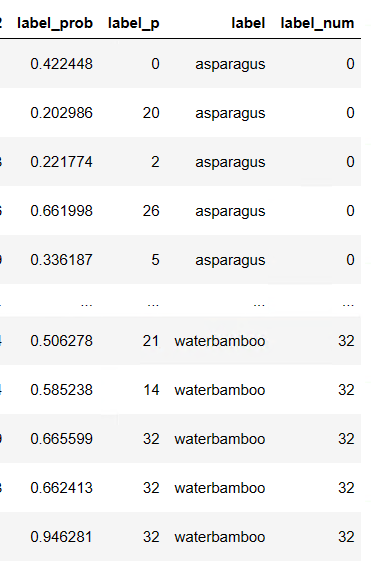
利用tensorflow.GradientTape()方法可以將模型指定的某一層抓出來，觀察資料到了這一層模型所注重的特徵位置，並利用熱像圖的方式顯示出來。利用Grad Cam可以確認模型是否學習到應該要學習到的特徵，也可以得知模型對於哪幾個類別會比較容易混淆，以便於後續的加強訓練。透過Figure 30可得知成果不如預期，因此利用Grad Cam分析訓練狀況也是值得改進的方向。



**Figure 30**：Grad Cam使用後失敗結果

4.投票:

原先我們預期能利用每張圖在不同模型中被預測出每個種類可能性的機率值，並透過權重的不同，將不同模型預測同種類之機率值相加取平均，找出機率值最高的種類作為預測結果。後期因時間因素，我們先假定權重相同做為測試，也因此造成模型對正確類別信心不足，進而被錯誤類別拉走。在未來改進方面上，或許找到各個模型最適合的權重並做加權，才能提升準確率。



**Figure 31**：投票結果(label\_prob:預測最大機率，label\_p:預測類別號碼

label:原圖類別，label\_num:原圖類別號碼)

|  |  |
| --- | --- |
|  | |

**程式碼**

https://drive.google.com/drive/folders/1o-HemWyh0CCPgH4PiRvHXxevfB6pbjg5?usp=share\_link

**使用的外部資源與參考文獻**

* Keras Applications

https://keras.io/api/applications/

**作者聯絡資料表**

* **隊伍**

| 隊伍名稱 | Private Leaderboard成績 | Private Leaderboard 名次 |
| --- | --- | --- |
| TEAM\_1927 | 0.8761583 | 40 |

* **隊員(隊長請填第一位)**

| 姓名 | 學校名稱 | 系所 | 電話 | E-mail |
| --- | --- | --- | --- | --- |
| 賴冠羽  (Guan-Yeu Lai) | 輔仁大學  Fu Jen University | 數學系資訊數學組  Information Mathematics Section in Mathematics Department | 0973-031-105 | yguanyappp23@gmail.com |
| 王譽錚  (Yu-Zheng Wang) | 輔仁大學  Fu Jen University | 數學系資訊數學組  Information Mathematics Section in Mathematics Department | 0919-606-252 | vivian20001122@gmail.com |
| 胡佩欣  (Pei-Hsin Hu) | 輔仁大學  Fu Jen University | 數學系資訊數學組  Information Mathematics Section in Mathematics Department | 0965-167-612 | gina50420@gmail.com |
| 龔意晴  (Yi-Ching Kung) | 輔仁大學  Fu Jen University | 數學系資訊數學組  Information Mathematics Section in Mathematics Department | 0988-193-409 | yichin11029@gmail.com |
| 何昱融  (Yu-Jung Ho) | 輔仁大學  Fu Jen University | 數學系資訊數學組  Information Mathematics Section in Mathematics Department | 0928-499-874 | malone.ho@gmail.com |

* **指導教授/指導業師**

| 教授/業師姓名 | 課程名稱 | 課程代碼 | 學校名稱 | 系所 | 電話 | E-mail |
| --- | --- | --- | --- | --- | --- | --- |
| 陳泓勳  (Hung-Hsun Chen) |  |  | 輔仁大學  Fu Jen University | 數學系  Mathematics Department |  | 152228@mail.fju.edu.tw |