AI 跨領域實作專題期末報告

不勞鵝獲的稻米

B10856012 吳明軒

B10856025 王郁晴

B10856050 蔡承恩

民國一一一年六月五日星期日

目錄

一、 問題與需求描述 1

二、 稻米分析 1

I. 資料分析 Analyze 1

II. 資料預處理 Process 1

III. 我們是如何使用模型的呢 ? What Model Do We Use? 2

IV. Model 模型功能呈現 2

V. Result 預測結果 3

三、 鵝隻異常行為分析 5

I. 資料分析 Analyze 5

II. 資料預處理 Process 6

III. Yolov5 鵝隻影像辨識 6

IV. Demo 9

四、 心得與分工 Experience 9

五、 總結: 9

一、 問題與需求描述

我們會想做這個是因為鵝隻會有異常現象，像是飛機翅、啄羽、軟腳...等等，所以我們想從到的飼料去做分析，偶然間就找到這一個 Dataset，所以我們就利用 **CNN** 進行分析，找出特徵，就可以對稻米的質量進行分類和評估。

稻米是世界上生產最廣泛的穀物產品之一，具有許多遺傳品種。由於它們的某些特徵，這些品種彼此分開。這些通常是紋理、形狀和顏色等特徵。有了這些區分稻米品種的特徵，就可以對種子的質量進行分類和評估。

那我們也做了一個能夠追蹤鵝啄羽的模型，所以以下會分為稻米分析及鵝隻異常行為進行講解。

二、 稻米分析

I. 資料分析 Analyze

在這項研究中，使用了 Arborio、Basmati、Ipsala、Jasmine 和 Karacadag 這五種常見於土耳其種植的不同稻米品種。

數據集中總共包含 75,000 張穀物圖像，其中每個品種有 15,000 張。

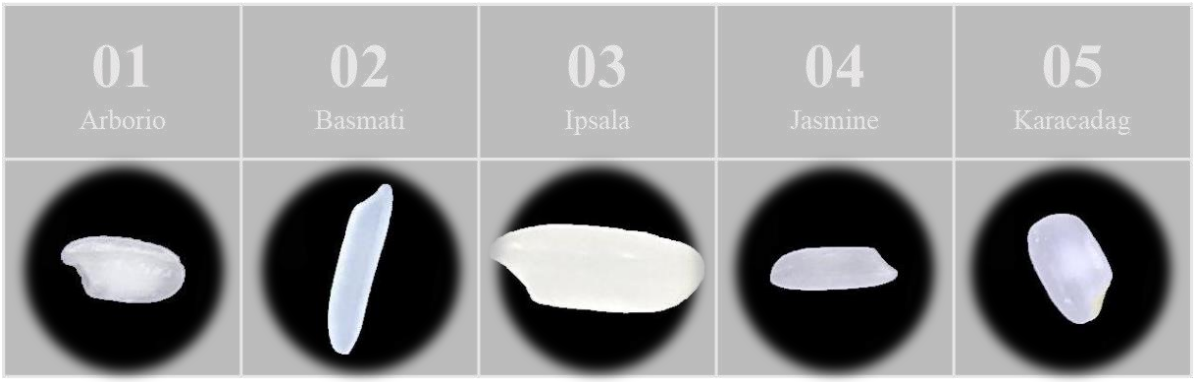

圖 1 稻米圖像

II. 資料預處理 Process

```
#批量修改檔名
path='Rice_Image_Dataset/Karacadag/'
files=os.listdir(path)

n=0
for i in files:
    oldname=path+files[n]
    n=n+1
    newname=path+'Karacadag_'+("%05d" % n)+'.jpg'
    os.rename(oldname,newname)
print('Done')
```

圖 2 批量修改檔案名稱

將原本檔名中因為有空白鍵等等因素，造成無法順利讀取檔案，所以我們特別將各個五種類別的稻米重新編號，從一編到一萬五千筆，總共七萬五千筆。

```
lables = ['Arborio', 'Basmati', 'Ipsala', 'Jasmine', 'Karacadag']
paths = 'Rice_Image_Dataset/'

X = []
Y = []
IMAGE_SIZE = 32

for label in range(len(lables)):
    for path in os.listdir(paths+lables[label]):
        img = np.asarray(load_img(paths+lables[label]+'/'+path, target_size=(IMAGE_SIZE,IMAGE_SIZE)))
        X.append(img)
        Y.append(np.asarray([label]))
        print(paths+lables[label]+'/'+path)
```

圖 3 改變圖片像素

下載圖片路徑的位置後，並將圖片從原本的 250*250 像素轉為 32*32 的像素，轉成陣列存到 X 裡。把標籤轉成陣列存到 Y 裡。

III. 我們是如何使用模型的呢 ? What Model Do We Use?

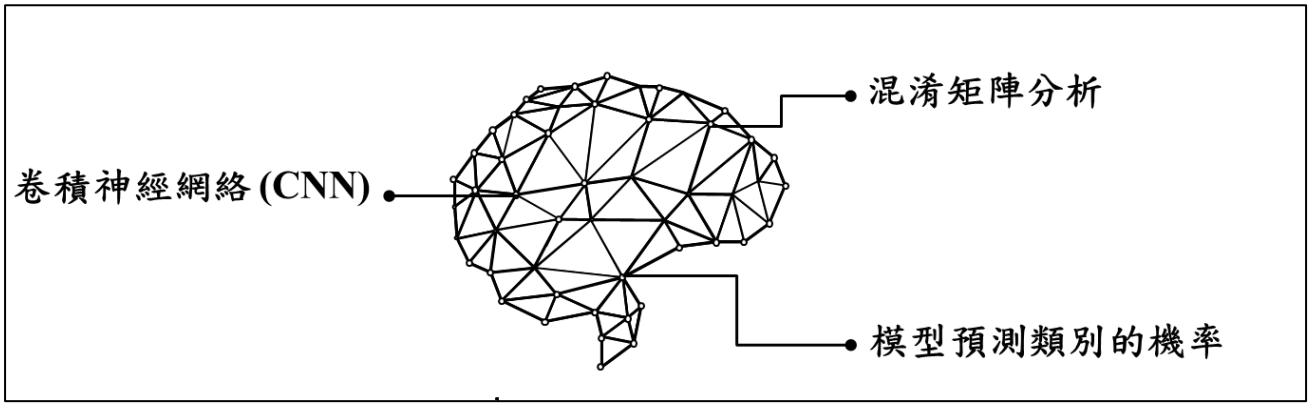

圖 4 模型分析流程圖

IV. Model 模型功能呈現

卷積神經網絡 (CNN)

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), padding="same",
                 input_shape=X_train.shape[1:], activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, kernel_size=(3, 3), padding="same",
                 activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(512, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(5, activation="softmax"))
model.summary()
```

圖 5 卷積層、平坦化、池化層、Dropout、 Dense

共有兩組捲積層和池化層、三組 Dropout、一個平坦層、兩個 Dense 層，啟動函數為 softmax。

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
dense (Dense)                (None, 512)               2097664
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 2565
=================================================================
Total params: 2,119,621
Trainable params: 2,119,621
Non-trainable params: 0
_________________________________________________________________
```

圖 6 每一層神經層的參數個數，和整個神經網路的參數總數

V. Result 預測結果

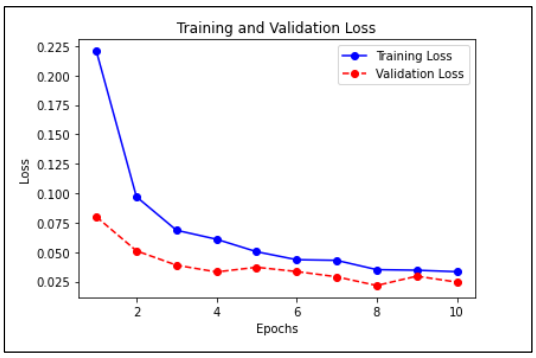

圖 7 訓練和驗證損失趨勢圖

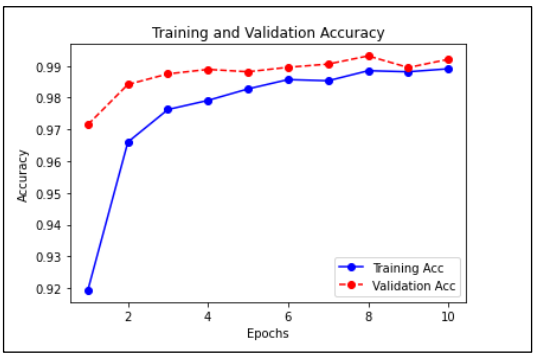

圖 8 訓練和驗證準確趨勢圖

```
Testing ...
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0195 - accuracy: 0.9938
訓練資料集的準確度 = 0.99
469/469 [==============================] - 2s 4ms/step - loss: 0.0271 - accuracy: 0.9921
測試資料集的準確度 = 0.99
```

圖 9 訓練及測試資料集準確度

由上方的兩張圖可見，驗證資料集準確率高達 0.99，測試集的準確度也是 0.99。以訓練和驗證損失的趨勢圖來看，線都越來越平穩，應該不會有過擬合的狀況發生。

混淆矩陣分析

評估分類結果的分析圖，每一列是真實標籤圖；每一欄是預測值。圖中從左上到右下對角線示預測正確的數量，落在其他區域的則是預測錯誤的數量。

predict	0	1	2	3	4
label					
0	2983	0	0	8	9
1	0	2955	0	45	0
2	5	0	2995	0	0
3	2	4	1	2993	0
4	24	0	0	0	2976

圖 10 混淆矩陣模型

模型預測類別的機率

I. 預測錯誤的類別:

我們可以看到下方這張稻米事[0，1，0，0，0]，1 在第二個位置，表示他原本是第二種類別稻米，但是很有可能把他誤判成下方長條圖 3 的位置，也就是第四種類別的稻米。而誤判的機率接近六成。

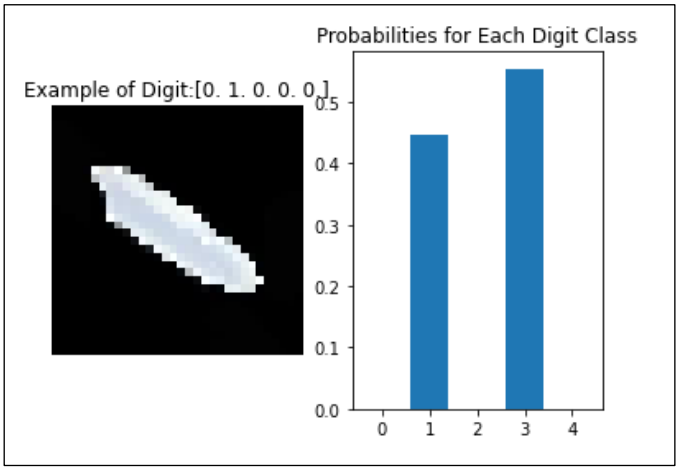

圖 11 預測錯誤

II. 預測正確的類別:

反之，下方的圖表示成功預測正確類別的稻米。

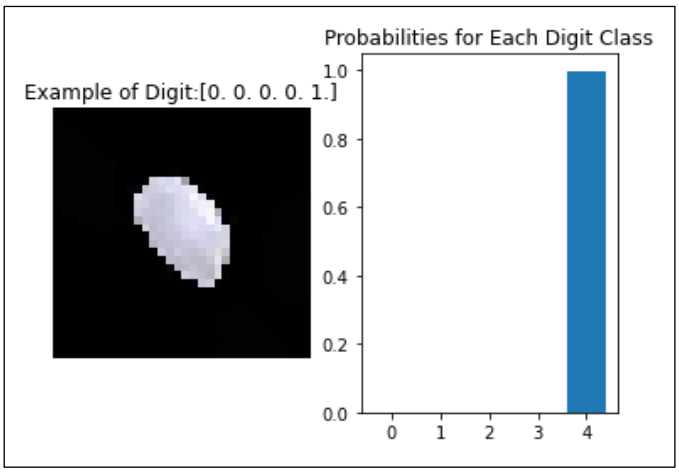

圖 12 預測正確

三、 鵝隻異常行為分析

本專題使用 Yolov5 進行物件偵測，抓取鵝隻邊框區域位置，再以 OpenCV 判斷鵝場是否有鵝隻有異常的啄羽行為。最後，透過 Deep Sort 技術，獲得鵝隻獨立中心點，與其不移動的持續時間，若停留超過指定時間，則判斷影像中的鵝隻有異常的啄與行為。

I. 資料分析 Analyze

共有 2839 張圖片作為分析的資料集

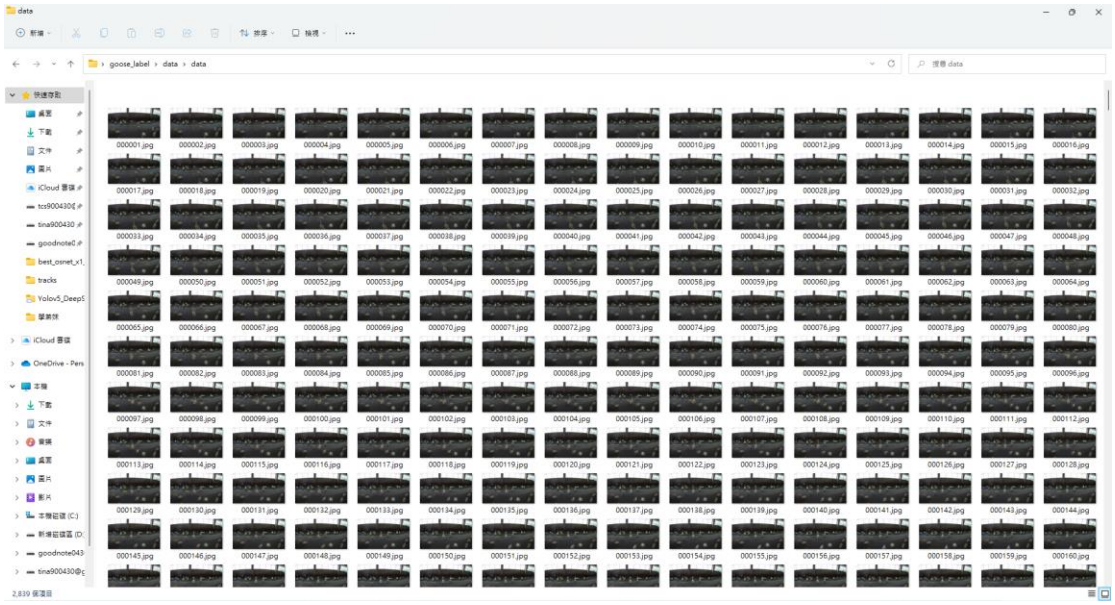

圖 13 本專題資料集

1. 事先架好的攝影機獲得大量的鵝隻影像將獲得的影像
2. 經過 python 做影像前處理，其中包括將影片拆成數幀、進行切割、擴增資料集等，進而得到大量的資料集。

II. 資料預處理 Process

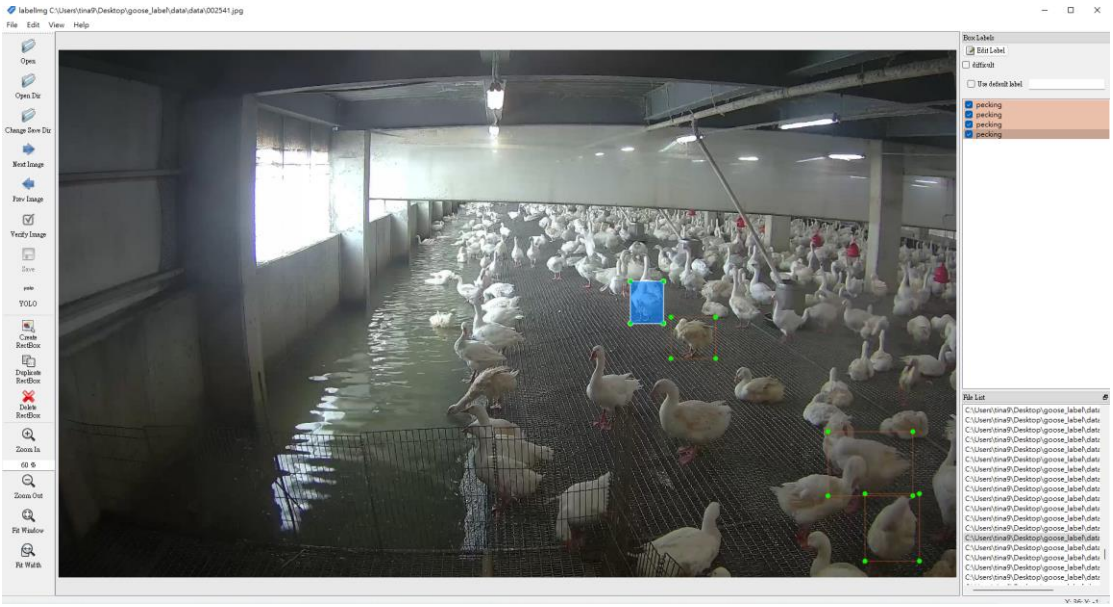

圖 14 標記的圖片

利用 Labelimg 將資料集進行人工標記，標記的標籤為 Pecking。

III. Yolov5 鵝隻影像辨識

本專題採用 Yolov5 深度學習演算法

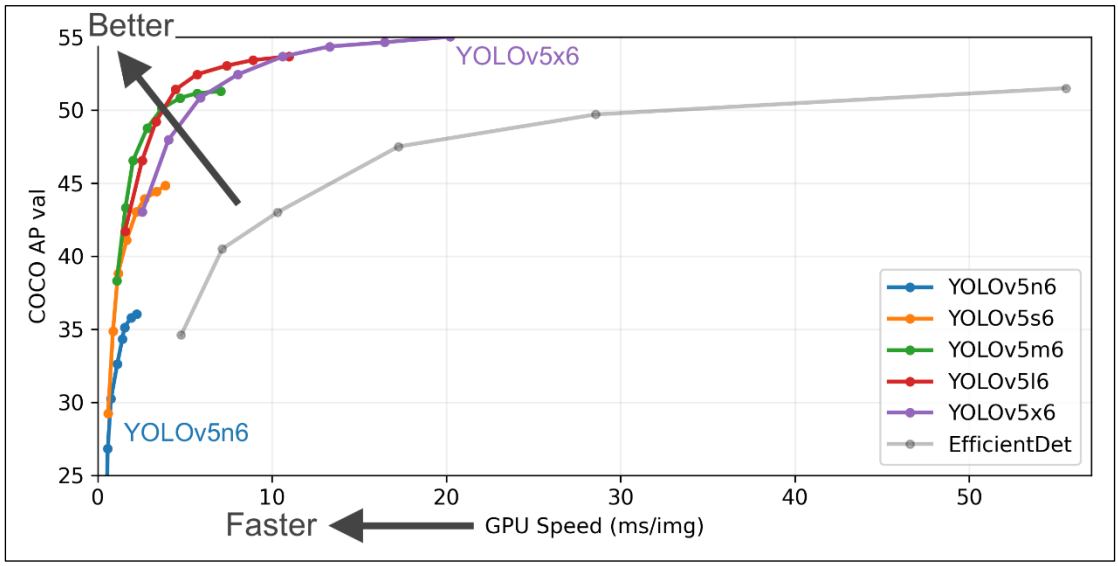

圖 15 Yolov5 各模型效能比較

Yolov5 數據增強

Yolov5 都會通過資料加載器傳遞每一批訓練數據，並同時增強訓練資料。資料加載器進行三種資料增強：縮放，色彩空間調整和馬賽克增強。下圖 17 是我們在訓練鵝隻數據時的數據增強結果。

圖 16 本專題訓練鵝隻數據時的數據增強結果

Activation Function

激活函數的選擇對於深度學習網路是至關重要的。Yolov5 的作者使用了 Leaky ReLU 和 Sigmoid 激活函數。在 Yolov5 中，中間/隱藏層使用了 Leaky ReLU 激活函數，最後的檢測層使用了 Sigmoid 形激活函數。

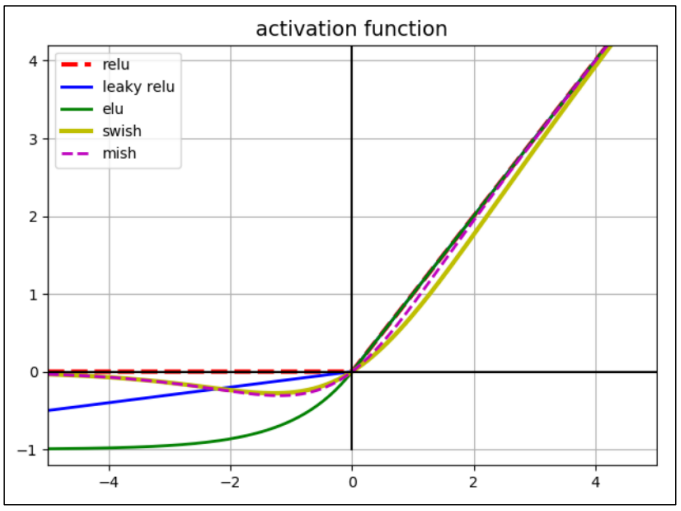

圖 17 激活函數

Optimization Function

Yolov5 的作者提供了兩個優化函數 Adam 和 SGD，並都預設了與之匹配的訓練超參數。默認為 SGD。而 Yolov4 使用 SGD。Yolov5 的作者建議是，如果需要訓練較小的自定義資料集，Adam 是更合適的選擇，儘管 Adam 的學習率通常比 SGD 低。但是如果訓練大型資料集，對於 Yolov5 來說 SGD 效果比 Adam 好。

Cost Function

Yolo 系列的損失計算是基於 objectness score, class probabilityscore,和 bounding box regression score。而 Yolov5 使用 GIOU Loss 作為 bounding box 的損失且利用二進制交叉熵和 Logits 損失函數計算類概率和目標得分的損失。同時我們也可以使用 fl _ gamma 參數來激活 Focal loss 計算損失函數。

將標記過後的資料集丟入 Yolov5 進行訓練，並觀察其訓練狀況，以便調整相關參數。

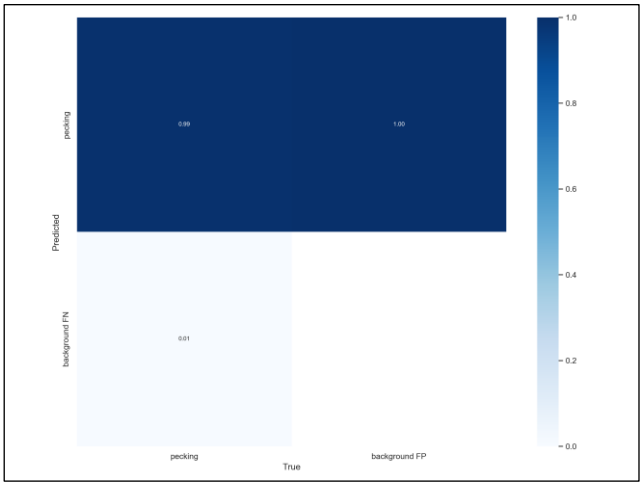

圖 18　Yolov5 模型的混淆矩陣(Confusion Matrix)

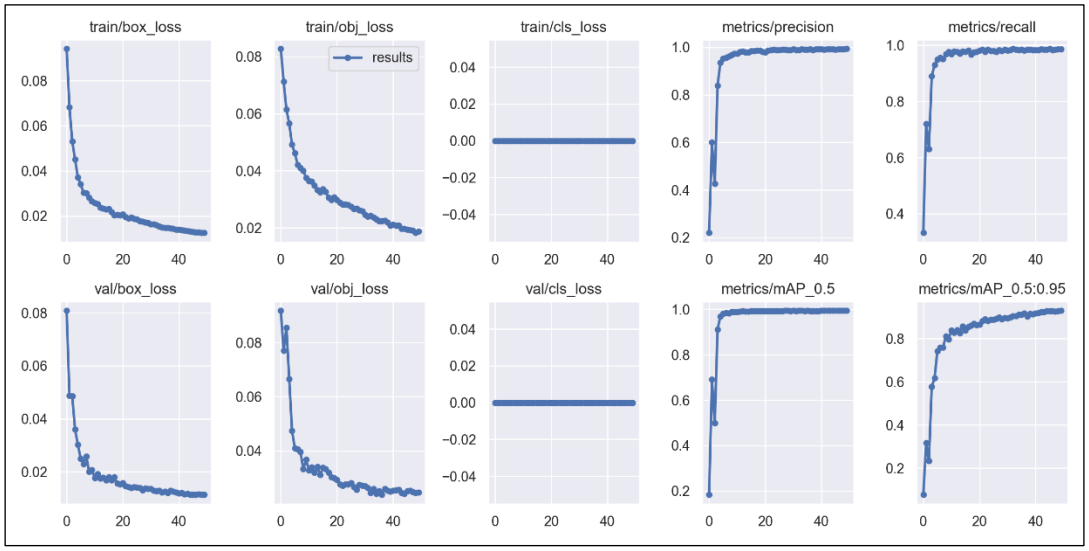

圖 19　Yolov5 最終訓練結果

利用 DeepSort 多目標跟踪踪演算法針對畫面中存在的物件進行排序與編號並且可得到一物件不離開畫面下其物體編號不變，因此本專題利用 Yolov5 得知物件座標，並利用 DeepSort 技術，判斷每物件中心並且個物件與中心點具有獨立性，以進行判斷鵝隻啄羽時長，進而得知是否有異常啄羽行為。

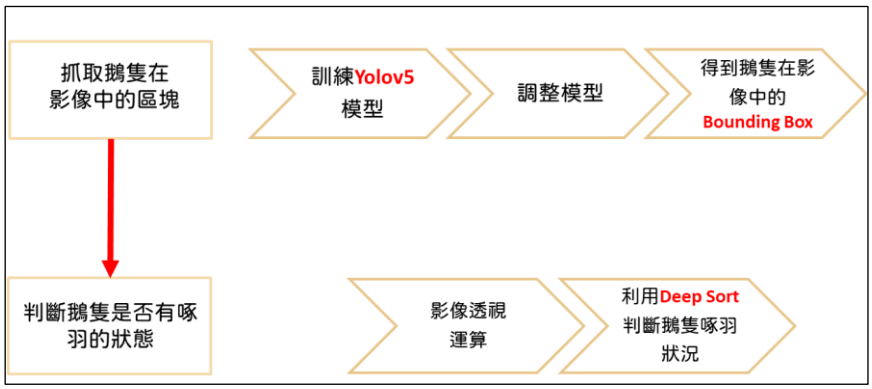

圖 20 Yolov5 最終訓練結果

IV. Demo

https://drive.google.com/file/d/1kVTkczA6Rtdt1aIeGj7iORzpvOOFMn_Q/view?usp=sharing

四、 心得與分工 Experience

	專題報告	程式撰寫	簡報製作	報告
吳明軒	✓	✓	✓	✓
王郁晴	✓	✓	✓	✓
蔡承恩	✓	✓	✓	✓

五、 總結:

透過這次的報告，我們把預測稻米類別與鵝隻行為做一個結合，將鵝隻的食衣住行中的食和行的照顧到，希望能把鵝隻養得更好。而這次的報告中，我們還尚未實行餵養鵝隻哪種類別的稻米會對鵝隻有更健康的行為，如果未來有機會的話，能夠真正去套用在鵝隻上來測試紀錄，或許就能更讓鵝隻減少異常行為。

B10856012 吳明軒心得: 經過這份報告，加深對於此堂課的應用，並且實作可本以外的有趣事物，透過嘗試錯誤，慢慢找尋問題，便能依依化解難題，將問題迎刃而解，藉著這次練習，讓深度學習的各個細節烙印於腦中，希望未來能有所應用，將其發揚光大。

B10856025 王郁晴心得: 藉由這堂課的報告實做一個有關 AI 預測的模型，其中找資料就花了好久的時間，以及資料預處理上也花費不少時間在處理，尤其是格式的問題。還有很重要的一點是電腦跑得快真的很重要!不然可能就會消耗一大段時間去跑程式而沒辦法充足的去更改更好的參數去訓練。

B10856050 蔡承恩：經過這份報告，我也將我這學期做的成果跟大家分享，在做的時候，剛開始花了一點時間研究，以及也花了很多人力幫忙 labeling。也經過這份報告，更加深了在課堂上上課的內容，在未來也更能夠將這些應用放進之後的專題報告裡面，做更專精的訓練預測。

參考資料：

I. https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset
II. https://github.com/ultralytics/yolov5
III. https://github.com/mikel-brostrom/Yolov5_DeepSort_OSNet
IV. https://github.com/KaiyangZhou/deep-person-reid/tree/e34e3ae85fe02314e3a9a9a93c5828f4fed1b225