

智慧化企業整合
Intelligent Integration of Enterprise
Project 3

利用 CNN 辨識水果名稱

109034549 溫芳苓

指導教授：邱銘傳 博士

目錄

一、研究背景	2
二、文獻回顧	3
三、研究過程	5
四、研究結果	8
五、結論與未來展望	15
六、參考資料	16

一、研究背景

1. 研究背景說明

現今水果業者大多採用人工結帳方式，其依水果的外型、顏色、表皮紋理等特徵來逐一辨識水果名稱，不僅耗時且可能辨識錯誤。而當水果攤販人潮眾多時，若消費者不了解某項水果名稱，其可能需耗費時間尋找或等待服務人員以取得資訊。因此，本研究欲透過 CNN 方法將九種水果（蘋果、酪梨、香蕉、櫻桃、芭樂、奇異果、檸檬、芒果、橘子）進行分類，並將本研究之模型提供予水果業者或消費者，以協助其辨識水果名稱。如此一來，本研究之模型不僅能提升水果業者結帳速度並降低人工辨識錯誤率，也能減少消費者尋找或等待服務人員所耗費的時間。本研究期望能為水果業者與消費者帶來便利性，進而提升水果業者之利潤。

2. 問題描述 (5W1H)

針對此研究主題，5W1H 分析說明如下表：

項目	內容
What	消費者於水果攤販可能不了解某水果名稱，而需尋找或等待服務人員以取得資訊；水果業者販售水果時需耗費時間逐一辨識水果名稱，且可能辨識錯誤造成財物損失。
Where	販售水果之店面（水果攤販）。
Who	欲了解某水果名稱之消費者、欲提升結帳速度或降低人工辨識錯誤率之水果業者。
When	當消費者欲了解某水果名稱時、水果業者欲提升結帳速度或降低人工辨識錯誤率時。
Why	建構水果辨識之模型，以減少消費者尋找或等待服務人員所耗費之時間，或提升水果業者結帳速度並降低人工辨識錯誤率。
How	九種水果圖片的資料前處理及 CNN 模型訓練，以協助消費者或水果業者辨識水果名稱。

二、文獻回顧

1. 卷積神經網路 (Convolutional Neural Network, CNN)

卷積神經網路 (Convolutional Neural Network, CNN) 是一種前饋神經網路，它的人工神經元可以回應一部分覆蓋範圍內的周圍單元，其對於大型圖像處理有出色表現。卷積神經網路由一個或多個卷積層和頂端的全連接層組成，同時也包括關聯權重和池化層 (pooling layer)，這一結構使得卷積神經網路能夠利用輸入資料的二維結構，且其在圖像和語音辨識方面能夠給出更好的結果。卷積神經網路需要考量的參數較少，使得其成為一種頗具吸引力的深度學習結構。

2. 卷積層 (Convolution Layer)

卷積層是一組平行的特徵圖 (feature map)，它透過在輸入圖像上滑動不同的卷積核，並執行一定的運算而組成。此外，在每一個滑動的位置上，卷積核與輸入圖像之間會執行一個元素對應乘積並求和的運算，以將感受野內的資訊投影到特徵圖中的一個元素。這一滑動的過程可稱為步幅 Z_s ，步幅 Z_s 是控制輸出特徵圖尺寸的一個因素。卷積核的尺寸要比輸入圖像小得多，且重疊或平行地作用於輸入圖像中，一張特徵圖中的所有元素都是透過一個卷積核計算得出的，也就是一張特徵圖共享了相同的權重和偏置項。

3. 池化層 (Pooling Layer)

池化 (Pooling) 是一種非線性形式的降採樣，有多種不同形式的非線性池化函式，而其中「最大池化 (Max pooling)」是最為常見的。它是將輸入的圖像劃分為若干個矩形區域，對每個子區域輸出最大值。

直覺上，這種機制能夠有效地原因在於一個特徵的精確位置遠不及它相對於其他特徵的粗略位置重要。池化層會不斷地減小資料的空間大小，因此參數的數量和計算量也會下降，這在一定程度上也控制了過擬合。一般來說，CNN 網路結構中的卷積層之間都會周期性地插入池化層，而池化操作提供了另一種形式的平移不變性。由於卷積核是一種特徵發現器，我們透過卷積層可以很容易地發現圖像中的各種邊緣。然而，卷積層發現的特徵往往過於精確，我們即使高速連拍拍攝一個物體，相片中的物體的邊緣像素位置也不大可能完全一致，而透過池化層我們可以降低卷積層對邊緣的敏感性。

4. 全連接層 (Fully Connected Layer)

在經過幾個卷積層和最大池化層之後，神經網路中最後的進階推理透過完全連接層來完成。就和常規的非卷積人工神經網路中一樣，完全連接層中的神經元與前一層中的所有啟用都有聯絡。因此，它們的啟用可以作為仿射變換來計算，也就是先乘以一個矩陣然後加上一個偏差 (bias) 偏移量 (向量加上一個固定的或者學習來的偏差量)。

三、研究過程

1. 資料說明及資料處理：

本研究利用 CNN 進行圖像辨識，且由 Kaggle 公開數據集中取得水果的圖像資料集，其中圖像皆為由 RGB 三原色所組成。其中包含九種水果：

- (1) 蘋果 (Apple) :656 筆圖像資料，其中 492 筆屬訓練集，164 筆屬測試集。
- (2) 酪梨 (Avocado) :570 筆圖像資料，其中 427 筆屬訓練集，143 筆屬測試集。
- (3) 香蕉 (Banana) :656 筆圖像資料，其中 490 筆屬訓練集，166 筆屬測試集。
- (4) 櫻桃 (Cherry) :656 筆圖像資料，其中 492 筆屬訓練集，164 筆屬測試集。
- (5) 芭樂 (Guava) :656 筆圖像資料，其中 490 筆屬訓練集，166 筆屬測試集。
- (6) 奇異果 (Kiwi) :622 筆圖像資料，其中 466 筆屬訓練集，156 筆屬測試集。
- (7) 檸檬 (Lemon) : 656 筆圖像資料，其中 492 筆屬訓練集，164 筆屬測試集。
- (8) 芒果 (Mango) : 656 筆圖像資料，其中 490 筆屬訓練集，166 筆屬測試集。
- (9) 橘子 (Orange) : 639 筆圖像資料，其中 479 筆屬訓練集，160 筆屬測試集。

九種水果共計 5,767 筆圖像資料，其中 4,318 筆屬訓練集，1,449 筆屬測試集。而本研究將訓練集中之 30% 設為驗證集。

```
from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val=train_test_split(X_train, Y_train, test_size=0.3, random_state=42)
```

本研究之圖像大小皆為 100*100，而本研究將圖像像素值標準化（即圖像像素值/255），以使其像素值從 0~255 等比例縮放至 0~1，且利用 One-hot encoding 將圖像（標籤）進行有效編碼。

```
#normalization and reshaping
X_train=X_train.reshape(-1, img_size, img_size, 3)
X_train=X_train/255
X_test=X_test.reshape(-1, img_size, img_size, 3)
X_test=X_test/255
print("shape of X_train= ", X_train.shape)
print("shape of X_test= ", X_test.shape)
```

```
shape of X_train= (4318, 100, 100, 3)
shape of X_test= (1449, 100, 100, 3)
```

```
from keras.utils import to_categorical
Y_train=to_categorical(Y_train, num_classes=9)
Y_test=to_categorical(Y_test, num_classes=9)
```

2. 資料擴增：

本研究利用 Data Augmentation 技術，將圖像進行縮放、移動、旋轉、翻轉，以避免過擬合情況發生，如下圖所示：

```
datagen=ImageDataGenerator(featurewise_center=False, #set input mean to 0
                           samplewise_center=False, #set each sample mean to 0
                           featurewise_std_normalization=False, #divide input datas to std
                           samplewise_std_normalization=False, #divide each datas to own std
                           zca_whitening=False, #dimension reduction
                           rotation_range=0.5, #rotate 5 degree
                           zoom_range=0.5, #zoom in-out 5%
                           width_shift_range=0.5, #shift 5%
                           height_shift_range=0.5,
                           horizontal_flip=False, #randomly flip images
                           vertical_flip=False,
                           )
datagen.fit(x_train)
```

Data Augmentation	參數設定
縮放	zoom_range=0.5
改變寬度	width_shift_range=0.5
改變高度	height_shift_range=0.5
旋轉	rotation_range=0.5
水平翻轉	horizontal_flip=False
垂直翻轉	vertical_flip=False

3. 建立模型：

本研究利用 Simple 2D CNN 來進行深度學習的模型訓練，本研究利用 Sequential 模型，其中包含兩層卷積層 (Convolution Layer)、兩層池化層 (Pooling Layer)、一層扁平層 (Flatten) 及兩層全連接層 (Dense Layer)。而為了避免過擬合情況發生，本研究亦加上 Dropout 層，架構如下圖所示：

```
model=Sequential()
model.add(Conv2D(filters=8, kernel_size=(3, 3), padding="Same", activation="relu", input_shape=(100, 100, 3)))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.35))

model.add(Conv2D(filters=16, kernel_size=(3, 3), padding="Same", activation="relu"))
model.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.35))

model.add(Flatten())
model.add(Dense(512, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(9, activation="softmax"))
```

由上圖可知，每一層卷積層之卷積核大小 (kernel_size) 皆為(3, 3)，激活函數皆為 relu，filters 則分別為 8、16；池化層之池化視窗大小 (pool_size)、步長 (strides) 皆為(2, 2)；Dropout 層設為 0.35、0.5；全連接層之激活函數分別為 relu、softmax。

```
#defining optimizer
optimizer=Adam(lr=0.0025, beta_1=0.9, beta_2=0.999)
#compile the model
model.compile(optimizer=optimizer, loss="categorical_crossentropy", metrics=["accuracy"])
model.summary()

epochs=10
batch_size=70
```

由上圖可知，本研究使用之優化器 (Optimizer) 為 Adam，學習率為 0.25%，損失函數為 categorical_crossentropy，迭代次數 (epochs) 為 10，batch_size 為 70。

四、研究結果

1. 模型校度分析

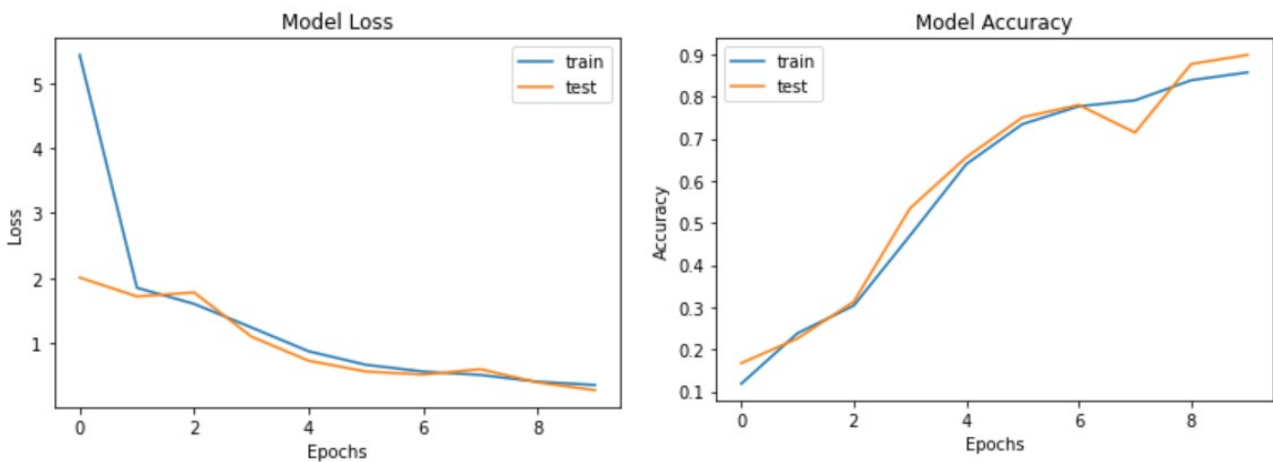
(1) 準確率

將本研究之模型進行校度分析，取得測試集準確率（test accuracy）為約 90%，如下圖所示：

```
train_loss, train_acc = model.evaluate(x_train, y_train, batch_size=batch_size)
test_loss, test_acc = model.evaluate(x_val, y_val, batch_size=batch_size)
print('Train accuracy:', train_acc)
print('Test accuracy:', test_acc)
```

```
44/44 [=====] - 6s 135ms/step - loss: 0.2369 - accuracy: 0.9136
19/19 [=====] - 3s 133ms/step - loss: 0.2680 - accuracy: 0.8989
Train accuracy: 0.9136333465576172
Test accuracy: 0.8989197611808777
```

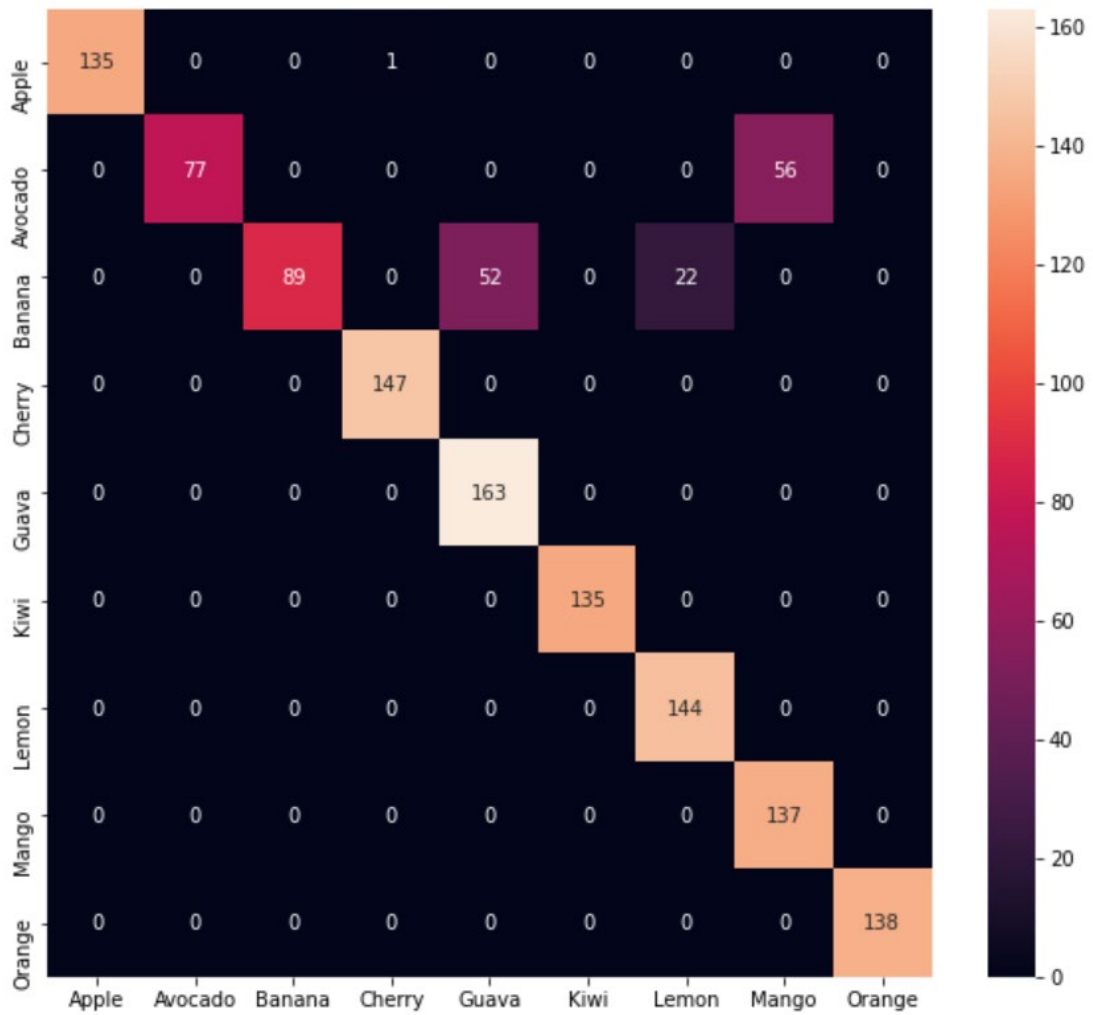
(2) 繪製 Loss、Accuracy 與 Epochs 之關係圖



由上圖 Loss、Accuracy 與 Epochs 之關係圖可知，學習越多回合其準確率會逐漸上升，Loss 則會逐漸下降，亦可發現有一點過擬合情況。

(3) 混淆矩陣 (Confusion Matrix)

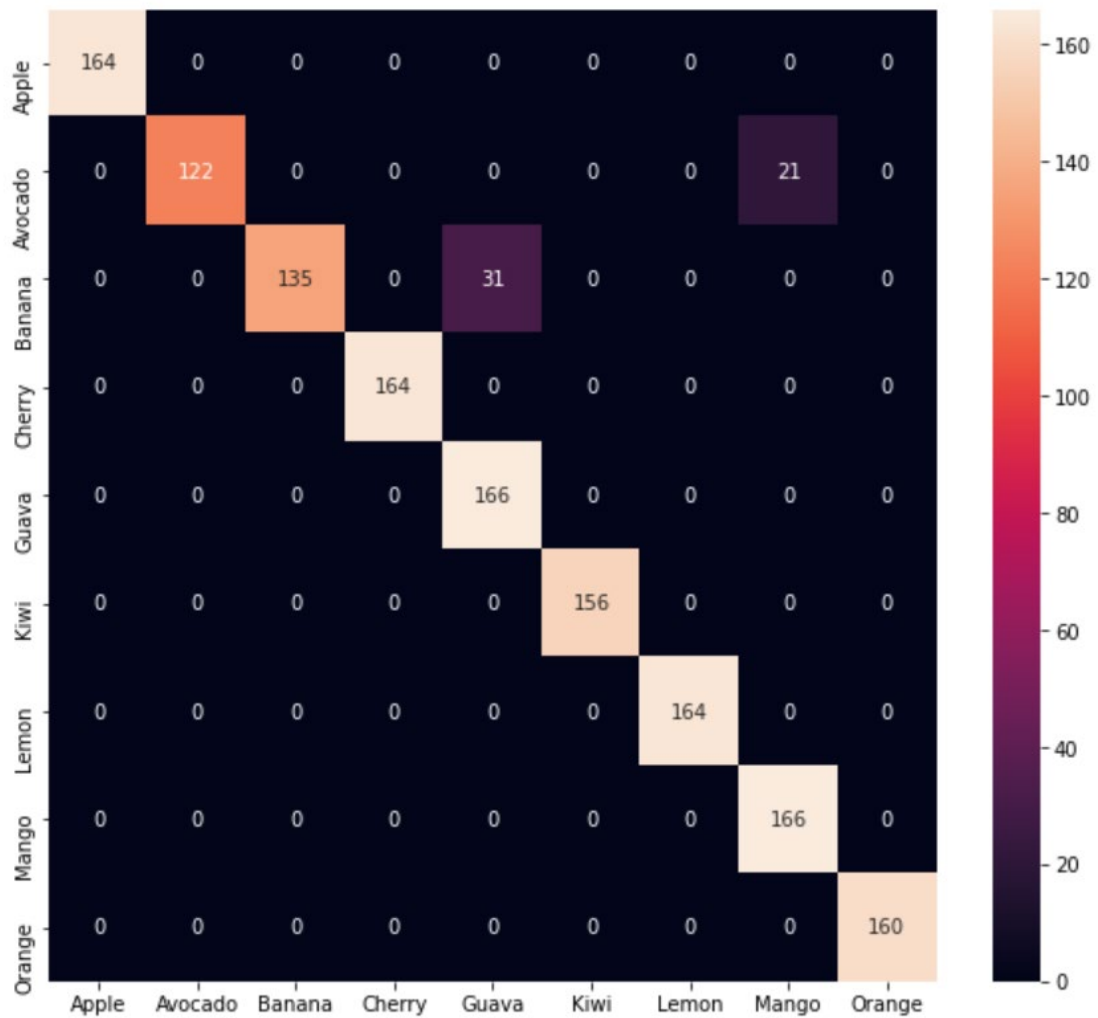
- 驗證集之混淆矩陣



(橫軸表預測水果名稱，縱軸表實際水果名稱)

由上圖可知，在 1,296 張驗證圖片中，有 1,165 張準確辨識，而有 131 張辨識錯誤，其中有 1 張蘋果圖片被辨識為櫻桃，有 56 張酪梨圖片被辨識為芒果，有 52 張香蕉圖片被辨識為芭樂，有 22 張香蕉圖片被辨識為檸檬。

● 測試集之混淆矩陣



(橫軸表預測水果名稱，縱軸表實際水果名稱)

由上圖可知，在 1,449 張測試圖片中，有 1,397 張準確辨識，而有 52 張辨識錯誤，其中有 21 張酪梨圖片被辨識為芒果，有 31 張香蕉圖片被辨識為芭樂。

2. 參數優化

本研究嘗試調整多個參數 (Batch_size、學習率、Layers、激活函數)，以取得更高之準確率。下表為本研究參數優化過程，且迭代次數 (Epochs) 為 10 固定不變：

	Batch size	學習率	Layers	激活函數	準確率
Origin	70	0.25%	兩層卷積層 (Filters 為 8、16)、 兩層池化層	relu	90%
Test 1	50	0.3%	三層卷積層 (Filters 為 8、16、32)、 三層池化層	relu	73%
Test 2	30	0.2%	四層卷積層 (Filters 為 8、16、32、64)、 四層池化層	relu	88%
Test 3	20	0.1%	三層卷積層 (Filters 為 8、16、32)、 三層池化層	sigmoid	53%
Test 4	15	0.1%	三層卷積層 (Filters 為 8、16、32)、 三層池化層	relu	99%
Test 5	15	0.05%	四層卷積層 (Filters 為 8、16、32、64)、 四層池化層	relu	80%

由上表可知，Test 4 參數設定 Batch_size 為 15、學習率為 0.1%、Layers 為三層卷積層 (Filters 為 8、16、32) 及三層池化層、激活函數為 relu 所得之準確率最高 (99%)。因此，本研究欲採用此參數設定來改善現有模型。

3. 參數優化結果

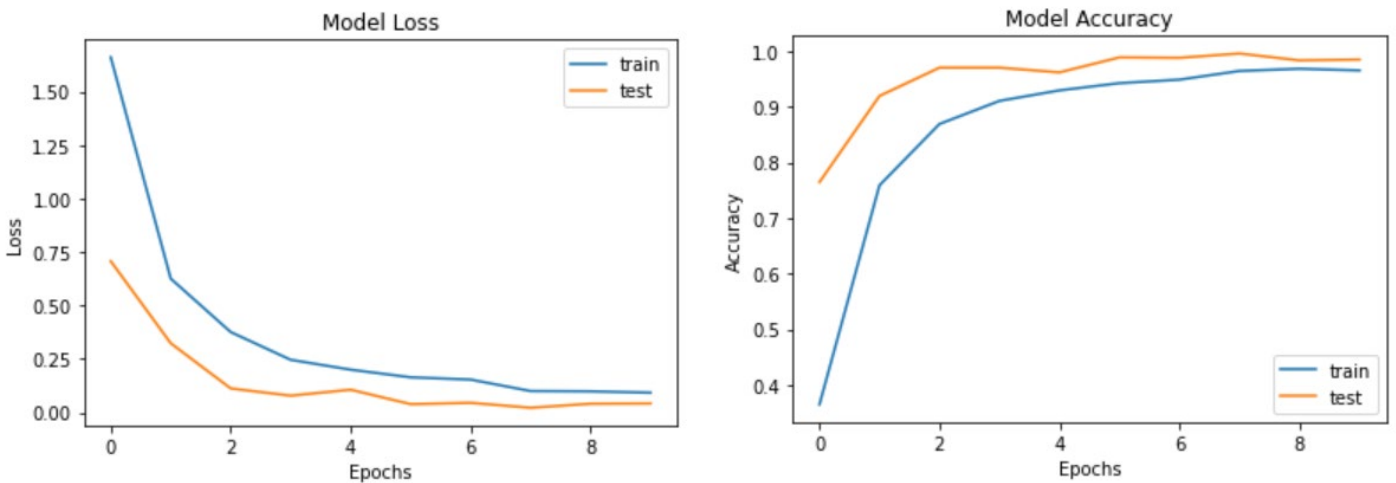
(1) 準確率

本研究採用 Test 4 參數設定來改善現有模型並進行校度分析，取得測試集準確率 (test accuracy) 為約 99%；與現有模型準確率相比提升了 9%。採用 Test 4 參數設定之準確率如下圖所示：

```
train_loss, train_acc = model.evaluate(x_train, y_train, batch_size=batch_size)
test_loss, test_acc = model.evaluate(x_val, y_val, batch_size=batch_size)
print('Train accuracy:', train_acc)
print('Test accuracy:', test_acc)
```

202/202 [=====] - 6s 31ms/step - loss: 0.0387 - accuracy: 0.9871
87/87 [=====] - 3s 31ms/step - loss: 0.0421 - accuracy: 0.9853
Train accuracy: 0.9870946407318115
Test accuracy: 0.9853395223617554

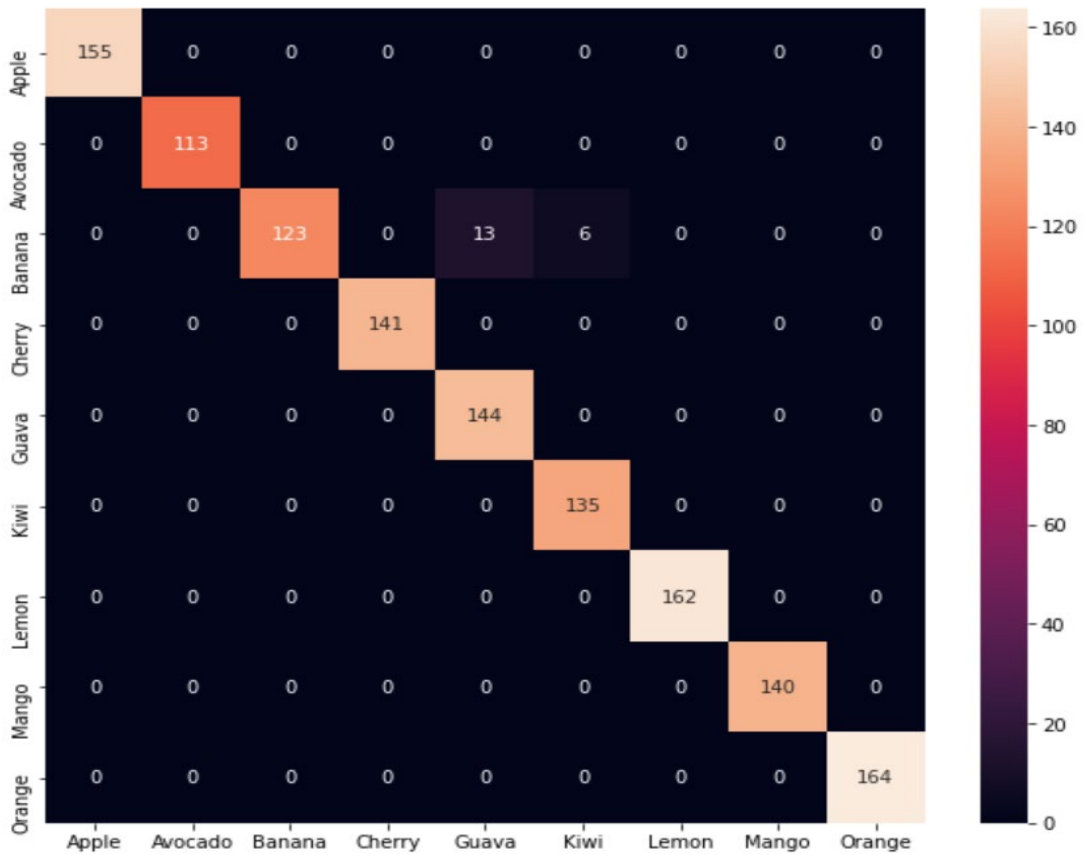
(2) 繪製 Loss、Accuracy 與 Epochs 之關係圖



由上圖 Loss、Accuracy 與 Epochs 之關係圖可知，學習越多回合其準確率會逐漸上升，Loss 則會逐漸下降，亦可發現並沒有太明顯之過擬合情況。

(3) 混淆矩陣 (Confusion Matrix)

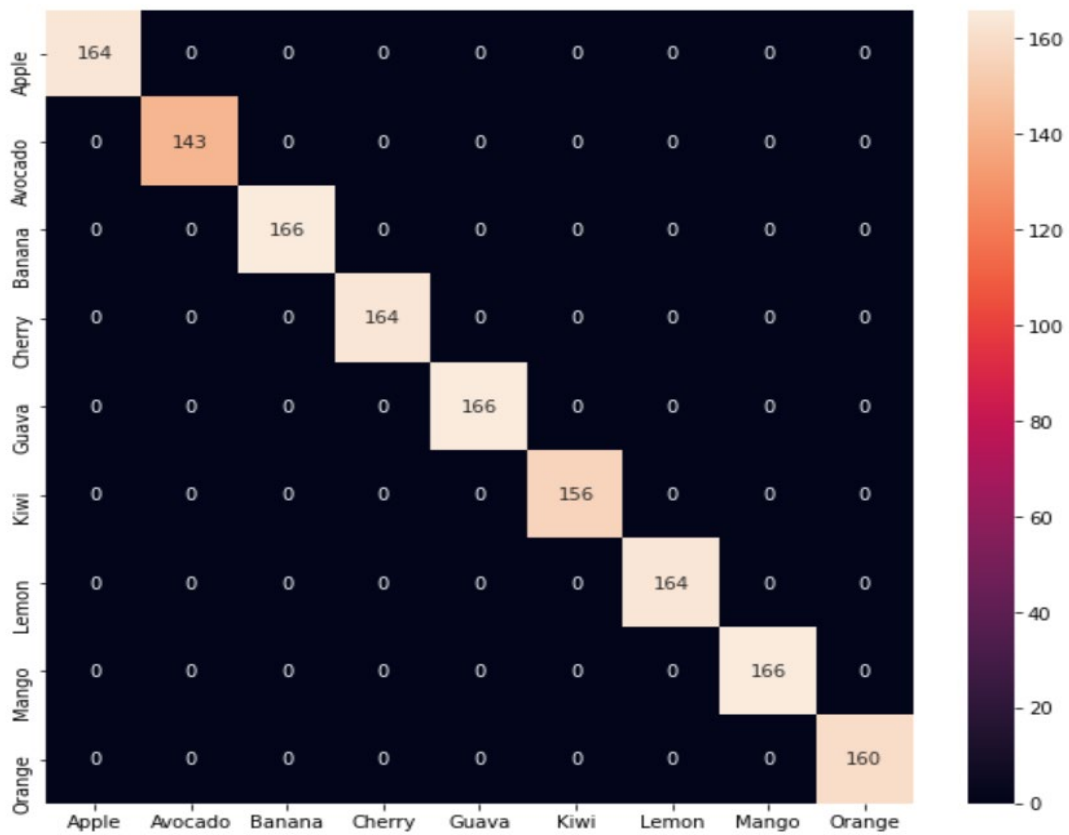
- 驗證集之混淆矩陣



(橫軸表預測水果名稱，縱軸表實際水果名稱)

由上圖可知，在 1,296 張驗證圖片中，有 1,277 張準確辨識，而有 19 張辨識錯誤，其中有 13 張香蕉圖片被辨識為芭樂，有 6 張香蕉圖片被辨識為奇異果。與現有模型之驗證集混淆矩陣相比，準確辨識之圖片增加了 112 張。

● 測試集之混淆矩陣



(橫軸表預測水果名稱，縱軸表實際水果名稱)

由上圖可知，1,449 張測試圖片皆準確辨識，沒有預測錯誤的情況發生，亦即預測結果與實際結果皆相同。與現有模型之測試集混淆矩陣相比，準確辨識之圖片增加了 52 張。

五、結論與未來展望

本研究利用 CNN 方法將九種水果（蘋果、酪梨、香蕉、櫻桃、芭樂、奇異果、檸檬、芒果、橘子）進行分類，且透過多次參數調整使得準確率可達 99%。本研究欲提供 CNN 模型予水果業者或消費者，以協助其辨識水果名稱。如此一來，本研究之模型不僅能提升水果業者結帳速度並降低人工辨識錯誤率，也能減少消費者尋找或等待服務人員所耗費的時間。然而，目前此模型僅針對九種水果進行研究，另外還有其他相當多種水果，且一種水果根據品種、熟度可能可再分為多種顏色、表皮紋理。因此，我認為未來能夠增加更豐富多樣之水果圖像並加以訓練，以建構一個完整的水果辨識模型。此外，我認為未來亦可利用不同的模型來訓練資料，並透過不同的資料集來驗證成果。CNN 可應用於相當多領域，未來可朝其他領域繼續研究。

六、參考資料

- <https://www.kaggle.com/etatbak/cnn-fruit-classification/data?select=fruits-360>
- <https://www.kaggle.com/moltean/fruits>
- <https://zh.wikipedia.org/wiki/%E5%8D%B7%E7%A7%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>
- <https://ithelp.ithome.com.tw/articles/10192028>