

智慧化企業整合

以 VGG-16 辨識鳥類

黃喆志 110034550

指導教授：邱銘傳

目錄

一、	前言.....	5
1-1	研究背景.....	5
1-2	5W1H.....	5
1-3	計畫流程.....	5
二、	資料蒐集與整理.....	6
2-1	資料來源.....	6
2-2	資料前處理.....	7
三、	模型建構.....	7
3-1	VGG16.....	7
3-2	模型建立與訓練.....	8
3-2-1	預訓練模型.....	8
3-2-2	添加圖層.....	9
四、	模型分析.....	9
4-1	確認 EPOCH 次數.....	9
4-2	參數優化.....	10
五、	結果與討論.....	11
5-1	泛化性測試.....	11
5-2	結論.....	12
5-3	未來展望.....	12

圖目錄

圖 1 計畫流程圖	6
圖 2 Bird Species Classification 資料集	6
圖 3 資料增強	7
圖 4 VGG16 示意圖	7
圖 5 VGG16 Summary.....	8
圖 6 預訓練模型程式碼 1.....	8
圖 7 預訓練模型程式碼 2.....	8
圖 8 預訓練模型程式碼 3.....	9
圖 9 VGG16 添加圖層後 Summary.....	9
圖 10 執行結果 1.....	10
圖 11 執行結果 2.....	10
圖 12 Mallard Duck.....	11
圖 13 泛化性測試程式碼.....	12
圖 14 class_dict.csv	12

表目錄

表 1 5W1H.....	5
表 2 實驗設計.....	10
表 3 L9 直交表結果.....	11

一. 前言

1-1 研究背景

鳥類為日常生活中最常見的脊索動物之一，不論是在附近的公園或是野外風景區都能見到其蹤跡，據調查台灣總共記錄到 87 科 674 種鳥類，但是大多數民眾對於野鳥認識甚少，不全然是對鳥類品種不感興趣，而是就算碰巧觀察到一隻特別的鳥想知道他是什麼名稱也僅能透過粗淺的描述，如毛色、體型大小……等關鍵字搜索，相當不便。

本研究以 VGG16 建立鳥類識別器，幫助使用者直接從影像就可以辨識鳥類。除了增進一般民眾對於野鳥的知識外，本研究結果也可以使用於學術單位或是與野鳥保育的相關單位，透過與野外攝影機的影像資料連結，及時且全天候的分析獲得的影像資料。本研究使用 Kaggle 上取得之鳥類圖像資料庫進行訓練，並藉此建立鳥類圖像分類。

1-2 5W1H

在分類鳥類時傳統的方式是透過外觀特徵來分類和查找，但是由於鳥類種類繁多，一般民眾賞鳥時若缺乏相關知識時難以分辨鳥類品種，賞鳥的行程即落為走馬看花。而以學術單位角度來看，若能將他們在他們研究野鳥生態的攝影機與此程式結合達到自動判斷攝影機影像中的鳥類，可以提升研究員的工作效率減輕研究員的負擔。

表 1 5W1H

Who	一般民眾、學術單位
What	鳥類品種辨識
When	需要辨識鳥類品種時
Where	鳥類出沒的地點
Why	鳥類種類繁多而且特徵難以辨識
How	深度學習、機器學習、資料分析

1-3 計畫流程

本研究蒐集 325 種鳥類的圖片，並針對所蒐集的圖片進行前處理，接著建構模型對其訓練，測試模型的訓練結果，藉由實驗設計流程不斷修正模型以達到參數優化提升準確度的目標，最後透過統計軟體 Minitab 找到最佳超參數調整組合並驗證之，計畫流程如圖 1 所示。

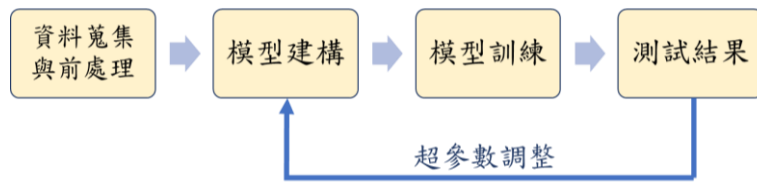


圖 1 計畫流程圖

二. 資料蒐集與整理

2-1 資料來源

本資料集使用 Kaggle 網站中的 325 Bird Species Classification 資料集，資料集包含 325 種鳥類照片，被劃分為訓練集 477332 張、驗證集以及測試集皆 1625 張。此外，所有圖像都是 jpg 格式的 224 X 224 X 3 彩色圖像，因此我們不需要格式化。此資料集有一大優勢為他已事先將所有數據拆分為訓練集、驗證集和測試集，幫助本研究能更聚焦於模型的訓練，並找出更好的模型。

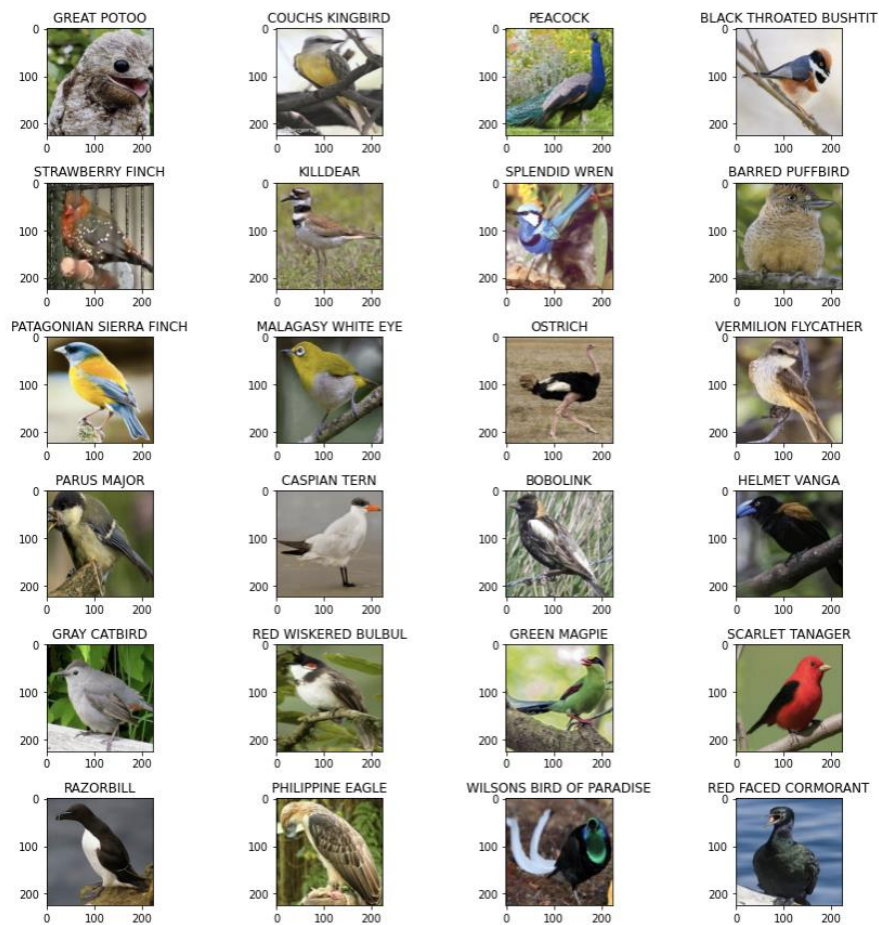


圖 2 Bird Species Classification 資料集

2-2 資料前處理

生成器在訓練深度學習模型時加載數據集。數據增強是一種從現有訓練數據人為地創建新數據的技術。它有助於：增加數據集的大小、在不使用額外數據的情況下在數據集中引入可變性。使用 ImageDataGenerator 來讓資料集之圖片進行標準化、裁切、縮放和水平翻轉，以利模型學習。其中“剪切”意味著圖像將沿軸扭曲，主要是為了創建或糾正感知角度，增強圖像以便讓電腦從不同角度所看所要訓練的圖片。

```
#Creating generator for Training DataSet
train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True, rescale=1/255)
```

圖 3 資料增強

三. 模型建構

3-1 VGG16

本研究使用之深度學習模型為 VGG16。VGG 是英國牛津大學 Visual Geometry Group 的縮寫，主要貢獻是使用更多的隱藏層，大量的圖片訓練，提高準確率至 90%。為 16 層(13 個卷積層及 3 個全連接層)，示意圖如圖 4.5 所示。

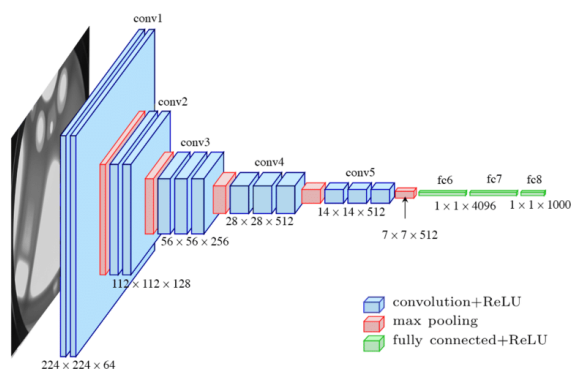


圖 4 VGG16 示意圖

```

Model: "vgg16"

```

Layer (type)	Output Shape	Param #
input_6 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

```

Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0

```

圖 5 VGG16 Summary

3-2 模型建立與訓練

3-2-1 預訓練模型

VGG16 模型從 ImageNet 上預訓練加載權重。VGG16 網絡的底層更接近圖像寬，而頂層更深。

```

base_model = keras.applications.vgg16.VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
base_model.summary()

```

圖 6 預訓練模型程式碼 1

凍結基礎模型中的所有層。冷凍層的優點是：

- 減少訓練模型時間
- 僅對幾個層進行反向傳播和更新權重，從而節省計算時間。

```

[24] base_model.trainable = False

```

圖 7 預訓練模型程式碼 2

3-2-2 添加圖層

圖 8 呈現的指令用來構建來自預訓練模型最後一層的新層。dropout 層可以防止模型過擬合。最後一個密集層中的 325 表示數據集中種類的總數。

```
#Create new model on top
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout
model=Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(2048,activation='relu',kernel_initializer='he_normal'))
model.add(Dropout(0.3))
model.add(Dense(2048,activation='relu',kernel_initializer='he_normal'))
model.add(Dropout(0.3))
model.add(Dense(325,activation='softmax',kernel_initializer='glorot_normal'))
model.summary()
```

圖 8 預訓練模型程式碼 3

```
Model: "sequential_2"
-----
```

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
flatten_2 (Flatten)	(None, 25088)	0
dense_6 (Dense)	(None, 2048)	51382272
dropout_4 (Dropout)	(None, 2048)	0
dense_7 (Dense)	(None, 2048)	4196352
dropout_5 (Dropout)	(None, 2048)	0
dense_8 (Dense)	(None, 325)	665925

```
-----
Total params: 70,959,237
Trainable params: 56,244,549
Non-trainable params: 14,714,688
-----
```

圖 9 VGG16 添加圖層後 Summary

四. 模型分析

4-1 確認 Epoch 次數

在尚未過度擬合的情況下，理論上 Epoch 次數越多準確率應越高，固本研究不將 Epoch 列為待調整的超參數，而是確認一個固定的 Epoch，往後的實驗設計都將套用這個 Epoch 數值。當吾人將 Epoch 設為 25 時，結果如圖 10.11 所示，可以發現兩張圖皆呈現收斂的狀態，故 4-2 節的實驗設計 Epoch 都將設為 25。

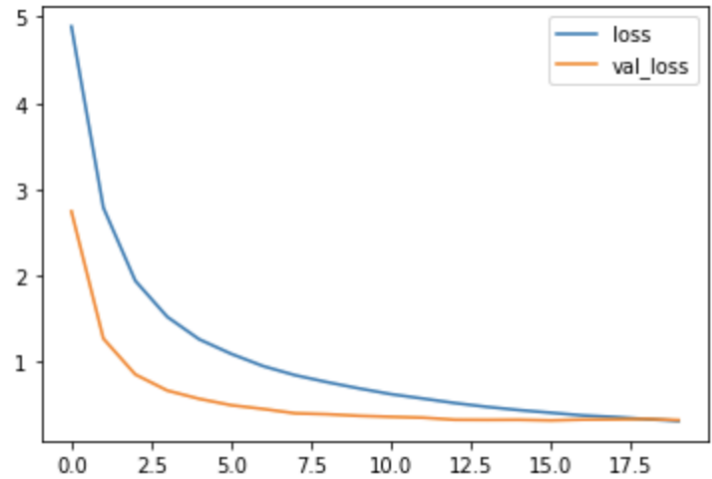


圖 10 執行結果 1

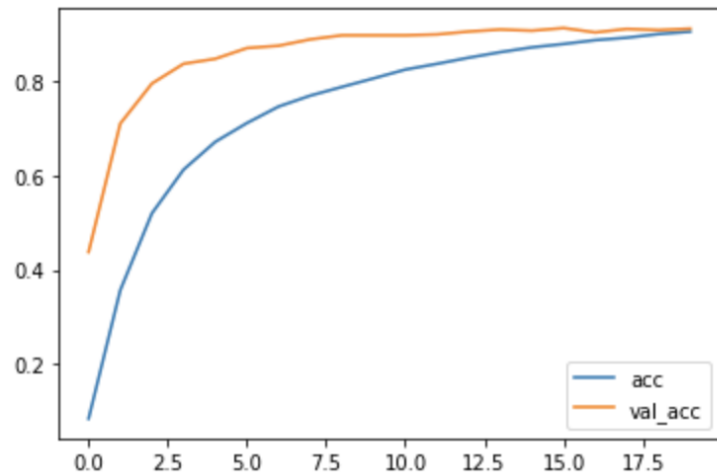


圖 11 執行結果 2

4-2 參數優化

本小組設計 3 因子 3 水準之實驗設計，其中選定之因子為 Dropout、Batch Size、Activate function。實驗設計之表格如表 2 所示：

表 2 實驗設計

Factor \ Level	Level 1	Level 2	Level 3
Dropout	0.3	0.4	0.5
Batch Size	32	64	128
Activate function	Relu	sigmoid	Tanh

表 3 L9 直交表結果

	Dropout	Batch Size	Activate function	train acc	test acc
1	0.3	32	Relu	0.9242	0.9206
2	0.3	64	sigmoid	0.9847	0.9348
3	0.3	128	Tanh	0.9876	0.9225
4	0.4	32	sigmoid	0.9748	0.9212
5	0.4	64	Tanh	0.9487	0.9274
6	0.4	128	Relu	0.8732	0.9206
7	0.5	32	Tanh	0.8535	0.9194
8	0.5	64	Relu	0.8874	0.9145
9	0.5	128	sigmoid	0.9455	0.9311

從表 3 實驗結果可以得知準確度最高的為實驗 2，準確度達到了 0.9348，將這 9 次實驗結果透過統計分析後可以得到各個因子對於準確率的影響程度，我們又從中挑選出了各個因子中最好的水準組合來做測試，希望可以得到更好的結果，最佳組合為 Dropout=0.3, Batch Size=64, Activate function 為 sigmoid，即實驗 2 是最佳組合。

五. 結果與討論

5-1 泛化性測試

吾人從資料集中隨意找一種鳥類品種並至 google 圖片搜尋此圖，誠如圖 12 中所示，此圖為 Mallard Duck。

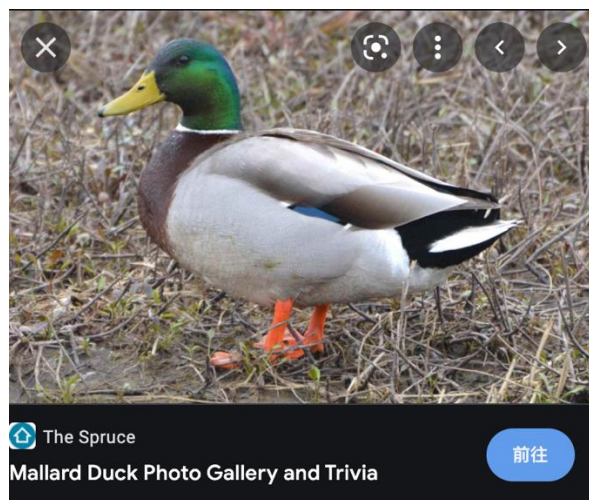


圖 12 Mallard Duck

```

from tensorflow.keras.preprocessing import image
from skimage import transform
img_width,img_height=224,224
img=image.load_img('/content/drive/MyDrive/mallard1-59511cd33df78cae81214312.jpeg')
img=image.img_to_array(img)
img=transform.resize(img,(224,224,3))
img=np.expand_dims(img,axis=0)
ans=model.predict(img)
print(ans.shape)
print(type(ans))

(1, 325)
<class 'numpy.ndarray'>

[22] print(np.argmax(ans))

188

```

圖 13 泛化性測試程式碼

圖 13 為泛化性測試程式碼示例，由於待泛化性圖片需要被讀進記憶體才能使用故使用 `Image.load_img` 這段 code 且需把圖片轉成陣列才能進行預測故使用 `image.img_to_array(img)`。此外，由於待泛化圖片需與預處理時的訓練圖片大小相同，故透過 `transform.resize(img,(224,224,3))` 執行。最後 `print(np.argmax(ans))`，誠如圖 14 所示，可以看出受試圖片被辨識為 index 第 188 項，也就是第 189 個類別。吾人從 kaggle 提供的 `class_dict.csv` 查找 index 第 188 項確實是 Mallard Duck，如圖 14 所示。可見泛化性測試是成功的。

187	MALEO
188	MALLARD DUCK
189	MANDRIN DUCK

圖 14 class_dict.csv

5-2 結論

本研究透過 VGG16 對 325 種鳥類進行分析，並得到良好的準確度得以見得 VGG16 優良的訓練成效，然而也因為較深的層數跟龐大的參數，訓練過程略顯冗長。在超參數優化環節，本研究透過實驗設計 L9 直交表並使用統計軟體統計出最佳組合，最後本研究用資料集以外的圖片進行辨識，且成功識別。

5-3 未來展望

本研究建構的鳥類辨識訓練模型在針對此一任務有不錯的準確度，然而端看這 325 種鳥類並無涵括台灣特有種，未來希望可以採用本土特有種鳥類的資料集進行訓練，讓本模型可以支援本土特有種鳥類的辨識。