

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

應用 CNN 模型辨識猴子種類

學生：邱韻婷 110034547

指導教授：邱銘傳 博士

中華民國 111 年 1 月 7 日

目錄

一、	研究背景.....	2
二、	問題定義.....	2
三、	文獻探討.....	3
	1. 影像辨識領域.....	3
	2. 卷積神經網路.....	3
四、	研究方法.....	5
	1. 資料來源.....	5
	2. 影像預處理.....	6
	3. 模型架構.....	7
	[1] VGG16.....	7
	[2] InceptionV3.....	8
	[3] Xception.....	9
	4. 超參數調整與模型比較.....	9
	[1] Adam :	10
	[2] SGD :	10
	[3] Adagrad :	10
五、	結論及未來展望.....	12
	1. 整體改善.....	12
	2. 未來展望.....	12

一、 研究背景

動物園是許多小朋友們最愛去的地方之一，那裡有各式各樣的動物種類，包括呆萌、懶惰、威武、兇猛的動物，甚至一種動物還分許多品種，常常讓人搞不清楚牠們的差異，而在動物園中最常辨識不出的就是猴子，猴子的種類繁多，動物園常見的有一般猴子、狒狒、獼猴、山魈及長臂猿等，牠們雖然都長得大同小異，但還是有一些較細微特徵的不同，舉例來說，川金絲猴（圖一）是台灣的國寶，只有台灣才有的一種猴子，又稱仰鼻猴，因為牠的鼻孔朝上，一身金燦燦的毛髮，為最漂亮的猴子；山魈（圖二）與猩猩的臉型差不多，但是牠的臉是彩色的，像京劇臉譜一樣，尤其是雄性山魈，臉上的顏色更醒目，以上都是不同品種猴子的差異與特徵。



圖一 川金絲猴



圖二 山魈

通常動物園會將不同種類的猴子安置在不同的觀賞區，遊客可透過旁邊的告示牌瞭解此區域的猴子為何種品種，然而，新竹動物園的猴子區劃分較不清楚，兩品種的猴子分別在同一區塊的上下位置，但告示牌沒有清楚表示只有圖片及對應的猴子名稱，因此往往造成遊客需比對多時或是詢問飼養員。為協助參觀動物園的大人孩童們可更快瞭解猴子種類，本研究將訓練 Kaggle：10 Monkey Species 資料集以準確辨識十種不同品種的猴子。

二、 問題定義

透過 5W1H 分析法，以協助我釐清問題及思考解決方法，為了使準確辨別各猴子品種，應用多種遷移學習結合 CNN 模型的方法，以比對不同方法間的差異及效果，本研究之 5W1H 如下表所示。

Why	協助父母於動物園時可以介紹不同種類的猴子與孩子互動，也供一般民眾對於猴子的認識
What	遊客可藉由本研究更快速辨識不同品種的猴子
Where	動物園或孩童學習環境中（家、學校、戶外等）
When	參觀猴子區或是讀猴子品種繪本時
Who	欲瞭解猴子品種的大人孩童
How	以深度學習 CNN 模型來訓練猴子品種資料庫

表一 5W1H 分析

三、 文獻探討

1. 影像辨識領域

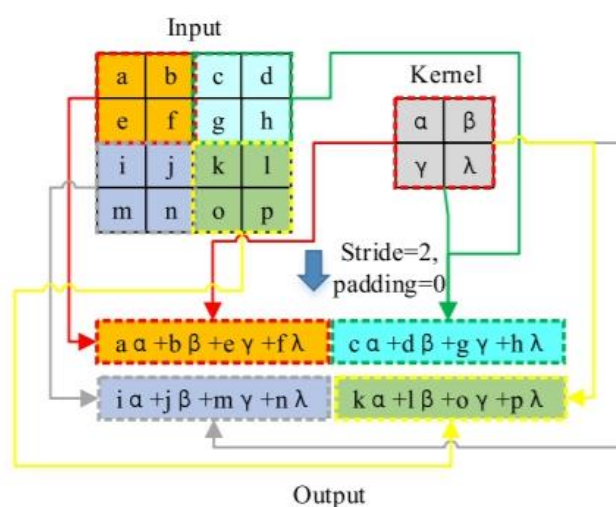
Fujiyoshi et al.(2019)整理出電腦視覺的應用包含 3 類，(1)物件偵測(Object Detection)：標註影像中某類別物體所在的位置及大小，應用於人臉辨識及行人檢測，類神經網路可以透過網路的訓練完成目標為多類別(兩類以上)的物件偵測(2)影像分類(Image classification)：將影像分類出所代表的類別，隨著類別增加，使用人眼辨識的準確度也隨之降低，因此此問題非常適合用於深度學習 (3)語意切割(Semantic Segmentation)：將影像按事物類別區分像素塊，利用傳統的機器學習很難解決影像中每個像素對各類別的語意切割，但是透過神經網路的學習可以很好地解決這個問題。

隨著人工智慧的發展，深度學習的發展越來越迅速，Wu(2017)提出深度學習結合許多層來解決任務或將抽象的數據進行分類，在不同領域中，可以將設計好的深度網路解決不同問題，不同網路有其各自的優勢，例如：長短期記憶(Long-Short Term Memory, LSTM)應用於自然語言處理、卷積神經網路(Convolutional Neural Network, CNN)應用於影像辨識，基於本次計畫的目的，以下針對卷積神經網路 CNN 做整理。

2. 卷積神經網路

卷積神經網路主要含有三種類型，分別是卷積層(Convolutional layer)、池化層(pooling layer)及全連接層(fully-connected layer)。

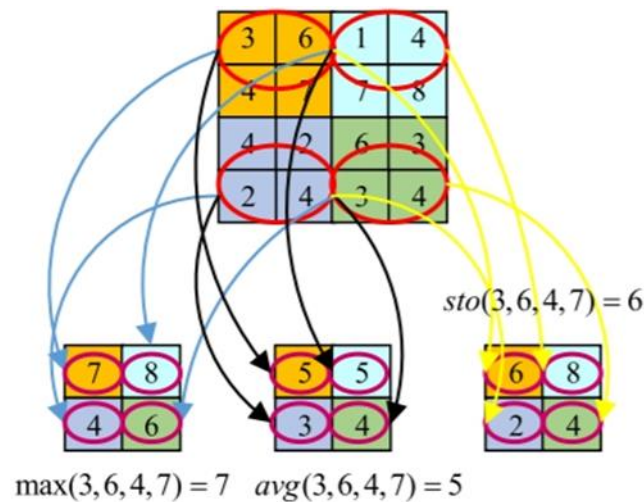
Liu(2018)卷積(Convolution)是通過特定運算符號對函數進行積分轉換，將原始函數改良成一個特定形式的兩個相似函數的新表示，被廣於應用影像處理以進行特徵萃取，而不會消除像素間的空間關聯，數學上，第一步透過行和列進行翻褶運算(稱卷積核，Kernel)建構單個卷積運算，第二步卷積核在原始圖片中從左上開始移動固定格數，第三步將原始圖片元素的值加權卷積核並相加，圖片位置會產生新值(圖三中， 2×2 矩陣即為新值)，有利於減少資料維度及產生較少的冗餘資料集，也被稱為特徵圖(Feature map)。卷積核是一種小矩陣可用於模糊、銳化、壓紋及邊緣檢測等，作為特徵辨識過濾掉原始圖片特徵存在的位置。



圖三 卷積運算的過程 資料來源：YOUHUI TIAN(2020)

由於直接分析原始圖片會需要進行大量的資料前處理，而卷積的宗旨在於自動萃取圖片的特徵，使電腦更可見，應用卷積不僅可以充分描述輸入的特徵，還可以降低手動操作的需求，低階卷積核的大小通常比原始圖片更小，但其特徵萃取關注在局部感知能力，高階卷積核可擴展至全領域，因此逐漸將局部特徵轉換成全部特徵，就像人類注視且辨別該圖片。

Guo et al.(2017)整理出池化層即採樣過程等同於模糊過濾，具有輔助萃取特徵的作用，可以減少特徵圖的尺寸並提高特徵萃取的穩定性，通常位於卷積層之間，根據核的移動步驟決定特徵圖的大小，典型的池化層設定是平均池化(Average pooling)即定義一個空間鄰域(例如：2x2的窗口)，並從窗口內的修正特徵圖算出平均值(圖四中，左下)及 Max pooling(最大池化)即從窗口內的修正特徵圖中取出最大的元素(圖四中，中下)，透過疊加卷積層和池化層可以萃取出更高階的特徵。



圖四 池化運算的過程 資料來源：YOUHUI TIAN(2020)

卷積神經網路的分類器是一個或多個全連接層，其獲取上一層所有神經元且將它們連接到此層的每個神經元，通常位於卷積神經網路的尾部，經過卷積層及池化層後產出的特徵，訓練數據集進行分類。除了以上三層外，當解決更複雜問題時，YOUHUI TIAN(2020)還會加入激活層(Activation layer)，激活函數會將非線性因素加進神經網路中，常用的激活函數包含 sigmoid 公式(1)、Tanh 公式(2)、ReLU 公式(3)等，將卷積層的線性計算結果進行了非線性映射，以形成更準確的分佈。

公式：

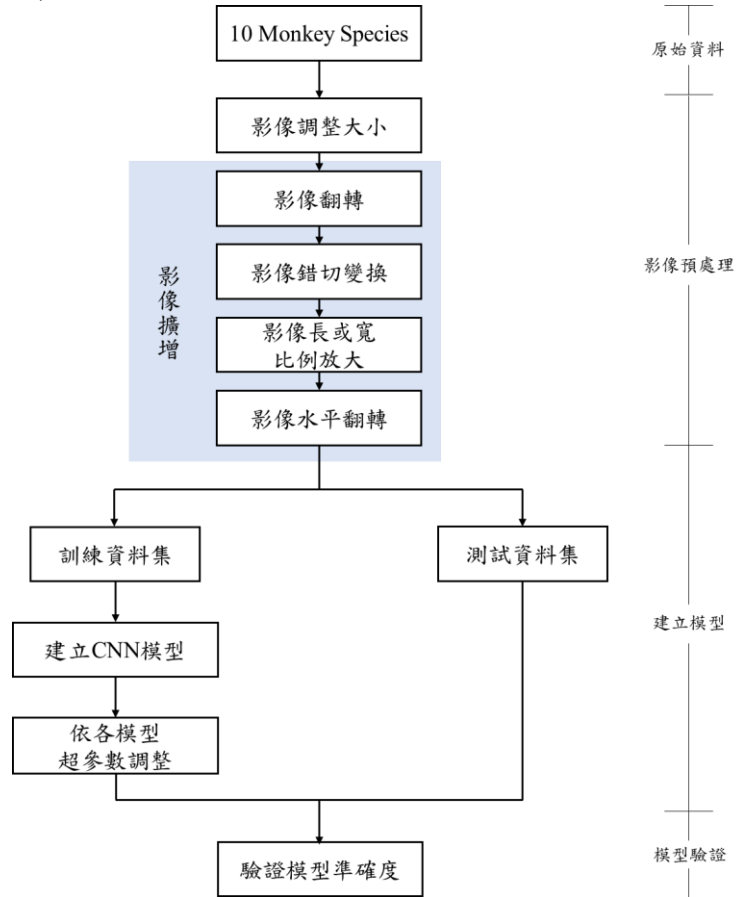
$$\sigma(x) = 1/(1 + e^{-x}) \tag{1}$$

$$\text{Tanh}(x) = (e^x - e^{-x})/(e^x + e^{-x}) \tag{2}$$

$$f(x) = \max(0, x) \tag{3}$$

四、 研究方法

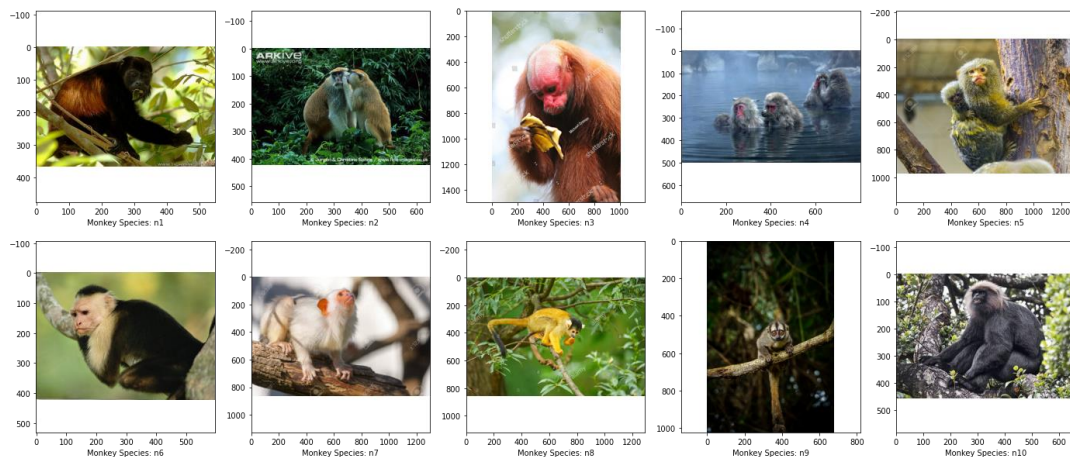
本研究之研究方法為資料準備(Data Preparation)、影像預處理(Pre-processing)、建立模型(Model Construction)、模型驗證與結果評估(Model validation and result evaluation)，將以上架構彙整成流程圖(圖五)。



圖五 研究方法流程圖

1. 資料來源

本研究之資料集來自 kaggle 網站所提供之 10 Monkey Species 資料庫，當中涵蓋測試資料集 1370 張影像（佔 83%）及驗證資料集 272 張影像（佔 17%），影像中包含 10 種不同品種（圖六）的猴子，影像皆為由 RGB 三原色所組成，如表二所示。測試資料集則由網路資源隨機選取以評量模型之泛化性。



圖六 資料集中 10 種猴子

表二 Kaggle 資料庫數據

Label	Latin Name	Common Name	Train Images	Validation Images
n0	alouatta_palliata	mantled_howler (鬃毛吼猴)	131	26
n1	erythrocebus_patas	patas_monkey (赤猴)	139	28
n2	cacajao_calvus	bald_uakari (白禿猴)	137	27
n3	macaca_fuscata	japanese_macaque (日本獼猴)	152	30
n4	cebuella_pygmea	pygmy_marmoset (倭猴)	131	26
n5	cebus_capucinus	white_headed_capuchin (巴拿馬白面卷尾猴)	141	28
n6	mico_argentatus	silvery_marmoset (狨猴)	132	26
n7	saimiri_sciureus	common_squirrel_monkey (松鼠猴)	142	28
n8	aotus_nigriceps	black_headed_night_monkey (黑夜猴)	133	27
n9	trachypithecus_johnii	nilgiri_langur (尼爾吉里葉猴)	132	26
合計			1370	272

2. 影像預處理

由於資料庫中的各影像尺寸大小皆不同，將所有影像調整成 224 X 224 同等大小 (影像 resize)，同時將影像像素值標準化處理，也就是將像素質同除 255，以使值從 0 至 255 等比例縮放至 0 至 1，提升資料訓練速度及準確度。

本研究利用資料擴增 (Data Augmentation) 之技術，將影像進行斜翻轉、錯切變換、水平翻轉、長寬比例放大等處理，以增加訓練資料集的數據，彌補訓練資料量不足的問題，同時提升模型訓練能力及增加泛化能力。如表三所示為所設定之參數。

表三 資料擴增參數

資料擴增項目	參數設定
rotation_range (指定旋轉角度範圍)	30
shear_range (錯切變換)	0.2
zoom_range (長及寬兩方向進行放大)	0.2
horizontal_flip (水平翻轉)	True

圖七為訓練資料集中的一張原始圖，圖八將該原始圖透過資料擴增所設定的參數輸出影像，除了尺寸不同外，還可觀察到以下參數設定與原圖間的差異，藉由隨機擴增資料庫影像以擴增訓練資料集。



圖七 原圖

rotation_range	shear_range	zoom_range	horizontal_flip

圖八 資料擴增影像輸出

3. 模型架構

本研究選定三種在各資料預測競賽中表現優良的 CNN 模型架構，分別是 VGG16、InceptionV3 與 Xception，藉由上述三種預訓練模型（Pre-Trained Model）以找出能預測出本資料集最佳結果的模型，並透過實驗設計的方式進行參數最佳化，以訓練出績效最好的模型。

[1] VGG16

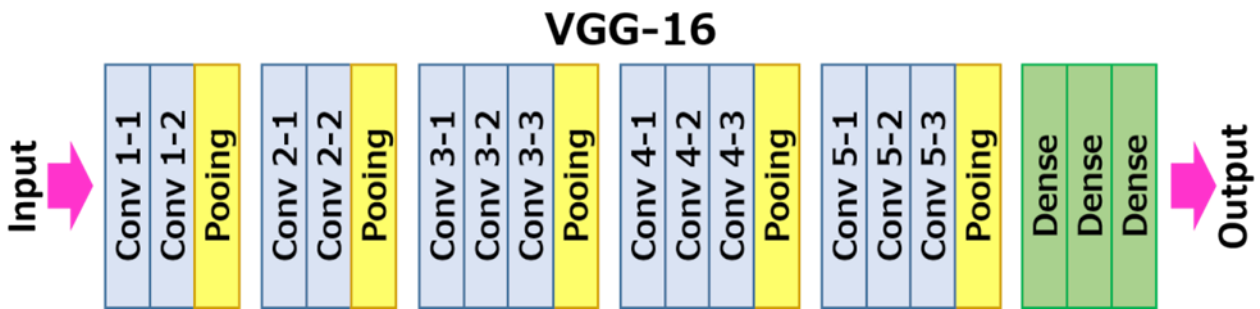
VGG 是英國牛津大學 Visual Geometry Group 的縮寫，主要貢獻是使用更多的隱藏層，以大量的圖片訓練，提高準確率至 90%。其結構簡潔，由 5 層卷積層、3 層全連接層、輸出層（激活函數 softmax）構成，層與層之間使用 Max-pooling（最大化池）分開，所有隱藏層的激活函數都採用 ReLU 函數。

VGG 應用多個較小卷積核（3x3）的卷積層代替一個較大卷積核的卷積層，一方面可以減少參數，另一方面相當於進行了更多的非線性映射，可以增加網絡的擬合及表達能力，由

於卷積核專注於擴大通道數、池化專注於縮小寬和高，使模型層數更深且特徵圖更寬，也控制計算量的增加規模，其缺點為參數量龐大，計算資源需求高，且較難調整參數。

16層包含13個卷積層及3個全連接層，其架構為卷積後接池化層，一來可減少參數的數量，加快計算速度，而減少參數可以防止過擬合的情況，且每個卷積都有一個激活函數，故可擬合更複雜的數據。

VGG16 架構圖如下，可觀察到其架構為每一塊包含若干卷積層和一個池化層，且同一塊內卷積層的通道（channel）數是相同的，隨著層數的增加通道數翻倍，圖片高和寬減半。

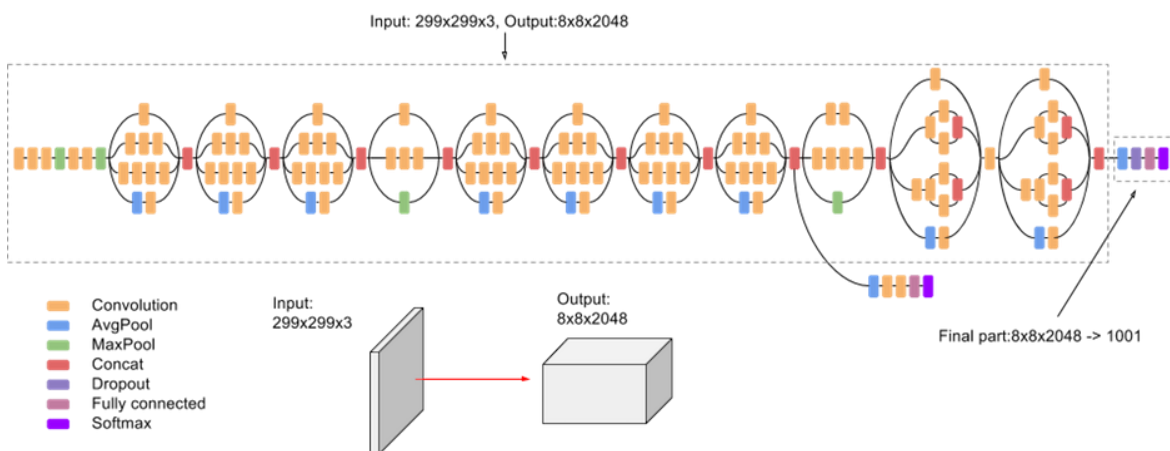


圖九 VGG16 架構圖

[2] InceptionV3

相較於 VGG16 其參數量較低且準度較高，Inception 為本架構的特色，使用較小的 filter 可以有效的降低所需的計算量，如 5X5 的 filter 計算量為 3X3 的 filter 的 25/9 倍，因此 Inception V3 可以實現快速的訓練，同時較寬的網路結構也可以避免產生瓶頸導致重要參數遺失，讓模型可更順利的與萃取特徵有關的參數。

Inception 具有許多版本，每個版本都是前一個版本的疊代進化，Inception v2 及 Inception v3 來自同一篇論文《Rethinking the Inception Architecture for Computer Vision》，作者提出了一系列能增加準確度和減少計算複雜度的修正方法，Inception v3 整合 Inception v2 中提到的所有外，還使用 RMSProp 優化器、Factorized 7x7 卷積、輔助分類器使用 BatchNorm、標籤平滑（增加至損失函式的一種正則化項，目的為防止過擬合情況）。

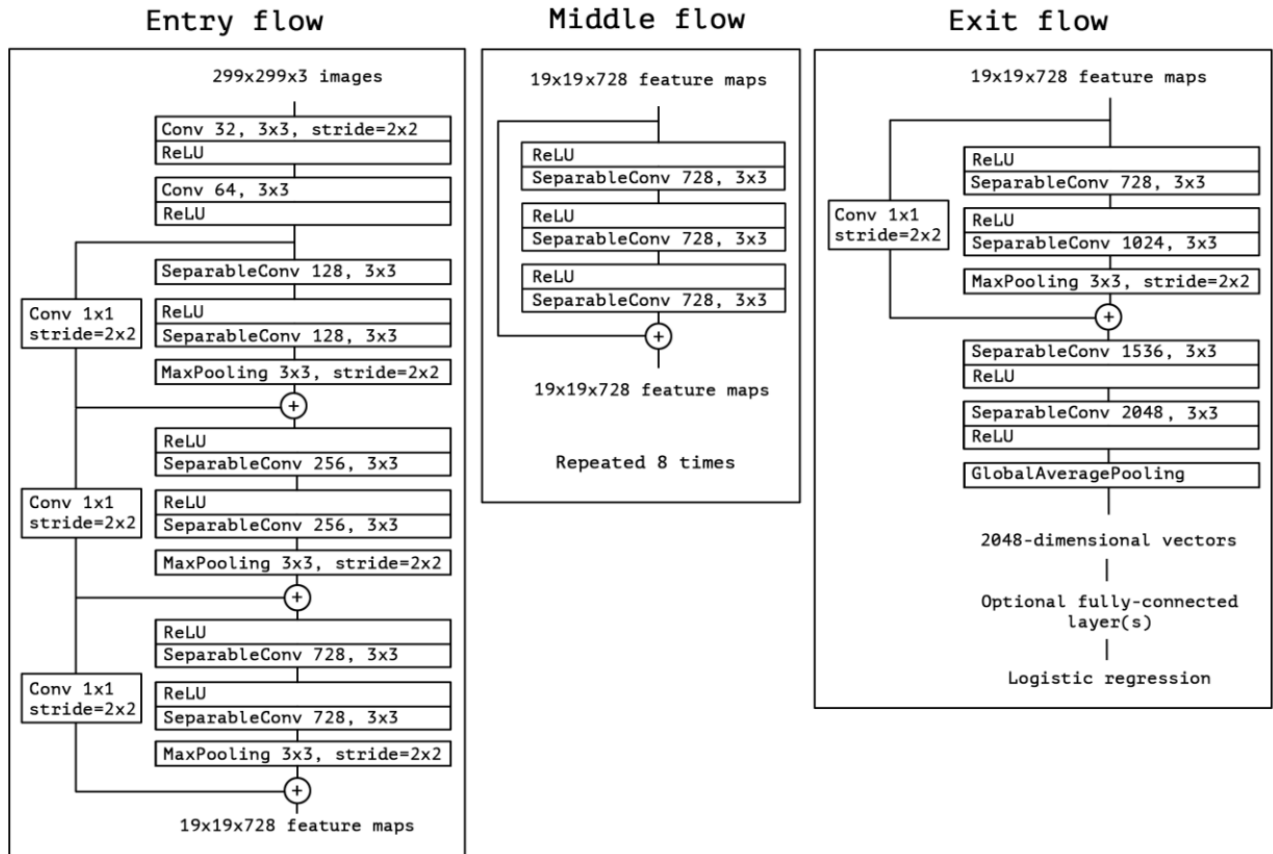


圖十 InceptionV3 架構圖

圖

[3] Xception

Xception 是指卷積神經網路的特徵圖中的跨通道相關性和空間相關性的對映，可以完全脫鉤，由於這種假設是 Inception 結構中極端化的假設，我們將它稱作 Xception，意指極端 Inception。Xception 在 ImageNet 上 top-1 的驗證準確度達 0.790，top-5 的驗證準確度達 0.945，需要注意的是輸入圖片的模式不同於 VGG16 和 ResNet 的 224x224，是 299x299，並且輸入預處理函式也有不同，另外目前該模型只支援 TensorFlow 作為後端。



圖十一 Xception 架構圖

4. 超參數調整與模型比較

除了上述三種 Pre-Train 模型，本研究選擇 Optimizer 及 Learning rate 作為預調整之超參數項目，如表四所示。

表四 超參數調整項目

Model	Optimizer	Learning rate
VGG19	SGD	0.01
InceptionV3	Adam	0.0001
Xception	Adagrad	

將上表中 Model、Optimizer（優化器）及 Learning rate（學習率）排列組合，完成本研究實驗設計，以找到準確率最高的模型，以下介紹所運用的優化器：

[1] Adam :

結合 AdaGrad 和 RMSProp 兩種優化演算法的優點，對梯度的一階矩估計（First Moment Estimation，即梯度的均值）和二階矩估計（Second Moment Estimation，即梯度的未中心化的方差）進行綜合考慮，其優點主要在於其有做偏置校正，使每次迭代（Epoch）學習率都有個確定範圍，讓參數的更新較為平穩，引數的更新不受梯度的伸縮變換影響，具有高校的運算速度且對記憶體需求少。

[2] SGD :

梯度下降法 (Gradient descent ,GD)是一次用全部訓練集的數據計算其損失函 數的梯度，即更新一次參數，由於每次都抽取相同的樣本集耗時且冗餘，而 隨機梯度下降法(Stochastic gradient descent, SGD)則是每次隨機抽取一個樣本或小批次(mini-batch)樣本即計算其梯度的平均後更新參數，由於隨機梯度下 降演算法每次只隨機選擇一個樣本來更新模型參數，因此每次的學習是非常快速的，並且可以進行線上更新。也會處於一個高變異的狀態，更新時 loss 比較震盪，可能會使得其跳出區域性最優點到達一個更好的區域性最優。

[3] Adagrad :

學習速度與參數會互相對應，舉例來說，頻繁常見的參數，那麼其會進行小部分的更新，反之，則會進行很大的更新，因此很適合處理稀疏資料集，且減少 Learning rate 的手動調整，但缺點就是分母會不斷積累，這樣學習率就會收縮並最終會變得非常小。

表五 其他參數設定

參數項目	數值
Epochs	10
Batch size	8
Loss	categorical_crossentropy

```
Xception_base = Xception(weights='imagenet',
                          include_top=False)

x = Xception_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(512, activation='relu')(x)

softmax_output_len = int(len(train_generator.class_indices.keys()))

predictions = layers.Dense(softmax_output_len, activation='softmax')(x)
Xception_transfer = models.Model(inputs=Xception_base.input, outputs=predictions)

Xception_transfer.summary()
optimizer1 = optimizers.SGD(learning_rate=0.01, momentum=0.9)
#optimizer1 = tf.keras.optimizers.Adagrad(learning_rate=0.01)
#optimizer1 = tf.keras.optimizers.Adam(learning_rate=0.01)
Xception_transfer.compile(loss='categorical_crossentropy',
                          optimizer=optimizer1,
                          metrics=['accuracy'])

history = Xception_transfer.fit(train_generator,
                               epochs=10,
                               shuffle = True,
                               verbose = 1,
                               validation_data = validation_generator)
```

圖十二 訓練模型程式碼（以 Pre-Train Model 為 Xception 為例）

表六為本研究實驗設計，可觀察到在準確度當中，粗體的儲存格是準確度最高的，在損失直當中，也代表損失是最低的。VGG19 測試資料集 (DOE2) 準確度最高的組合：優化器 SGD、最低學習率設置為 0.0001；Xception 測試資料集 (DOE10) 準確度最高的組合：優化器 Adam、最低學習率設置為 0.0001；Inception V3 測試資料集 (DOE16) 準確度最高的組合：優化器 Adam、最低學習率設置為 0.0001

表六 實驗設計—模型驗證結果

	Model	Optimizer	Learning rate	Val_Acc	Val_loss
1	VGG19	SGD	0.01	0.9063	0.5988
2			0.0001	0.929	0.421
3		Adam	0.01	0.9143	0.6274
4			0.0001	0.9215	0.4537
5			0.01	0.9021	0.542
6			0.0001	0.8879	0.7034
7	Xception	SGD	0.01	0.9243	0.1453
8			0.0001	0.9421	0.842
9		Adam	0.01	0.9618	0.1379
10			0.0001	0.985	0.0314
11			0.01	0.9387	0.2391
12			0.0001	0.9381	0.4614
13	InceptionV3	SGD	0.01	0.9763	0.071
14			0.0001	0.9599	0.2009
15		Adam	0.01	0.9362	0.2653
16			0.0001	0.9757	0.0367
17			0.01	0.9872	0.0412
18			0.0001	0.9135	0.6036

分別將各模型準確度高的訓練過程陳列至下圖，左右邊分別為訓練資料及驗證資料的 Loss、與準確度，可觀察到雖然訓練資料集 Loss 會不斷下降，但驗證資料集隨著 Epoch 增加漸漸地收斂，而準確度的部分可以觀察到訓練資料集跟驗證資料集的準確率都持續上升，沒有發生只有訓練資料準確度持續上升但驗證資料準確度沒上升的情況，也證實本組的模型沒有過擬合。

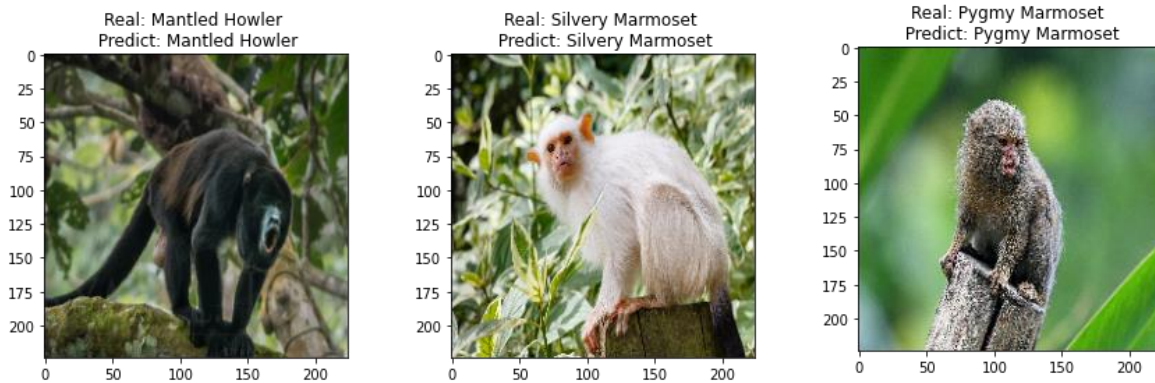


圖十三 最佳 Xception 模型訓練結果



圖十四 最佳 Inception V3 模型訓練結果

透過網路上蒐集測試資料以驗證本模型除具有高準確率外，也具有泛化能力，可觀察到皆可準確預測出其品種。



圖十五 測試資料集辨識結果

五、 結論及未來展望

1. 整體改善

從模型結果可以得出，嘗試上述不同模型組合後，本研究在 Xception 的模型表現最佳，訓練準確率能達 0.99，驗證的準確率也高達 0.99，透過遷移學習的方式讓模型訓練時間更短、收斂速度更快及更精準的權重參數。而在實際測試中真的將網路上任一猴子種類上傳本模型也可準確辨識種類，由此更展現本研究之模型具有泛化能力。

2. 未來展望

未來若是能將此模型結合網站平台可發揮更大效益，當遊客參觀動物園時可真正用手機拍現場猴子以辨識其品種，甚至，動物園也可結合此網站開發 APP 結合行銷方案及遊園活動，以吸引更多遊客入園參觀。

參考資料

- [1] Y. TIAN.(2020).“Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm,” IEEE Access,8: 125731 – 125744.
- [2] H. Fujiyoshi, T. Hirakawa, T. Yamashita.(2019).“Deep learning-based image recognition for autonomous driving,” IATSS Research,43:244–252.
- [3] <https://kknews.cc/zh-hk/n/e58ma3n.html>
- [4] <https://cinnamonaitaiwan.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-cnn%E5%8E%9F%E7%90%86-keras%E5%AF%A6%E7%8F%BE-432fd9ea4935>
- [5] <https://www.itread01.com/content/1548669068.html>
- [6] <https://www.itread01.com/content/1549635124.html>