中華大學生物資訊學系系統開發專題報告

台灣常見螞蟻之深度學習影像辨識

Deep learning image recognition of common ants in Taiwan

專題組員: 陳紘睿、洪愷俐、吳佩琦

專題編號：PROJ-BIOINFO-104004

指導老師: 董其樺老師

1.摘要

本專題與螞蟻帝國合作，利用類神經網路之深度學習打造一個可辨識台灣本土特有五種螞蟻的影像辨識模型，可讓螞蟻帝國與顧客溝通並解決客人的問題。我們採用卷積神經網路的架構，將螞蟻帝國提供的照片裁剪後，進行一系列參數組合測試。從 13 種組合測試中挑選出準確率較高且無過度擬合的模型。專題最終成果為，台灣特有五種螞蟻的影像辨識率最高可達八成。

2.簡介

近幾年機器學習開始蓬勃發展，許多社會問題開始用到機器學習的方法來解釋或預測，例如過去鐵達尼號旅客資料集完整保留下來後，透過類神經網路的學習，針對乘客的個人資料與相關資訊來預測他們的存活率。這樣的資料集已經被做為類神經網路入門學習的範例[1]。

人工智慧開始於 1950 年代，此領域發展的目的是希望能讓電腦像人一般思考與學習。機器學習是人工智慧的分支，透過演算法使用大量資料進行訓練後產生模型。往後當有新的資料出現，便可使用訓練產生的模型來進行分類分群或預測推估。而深度學習又是機器學習領域最新的發展，是人工智慧中成長最快的領域，主要是透過模擬人類神經網路的運作方式，進而學習龐大資料的內涵[2]。

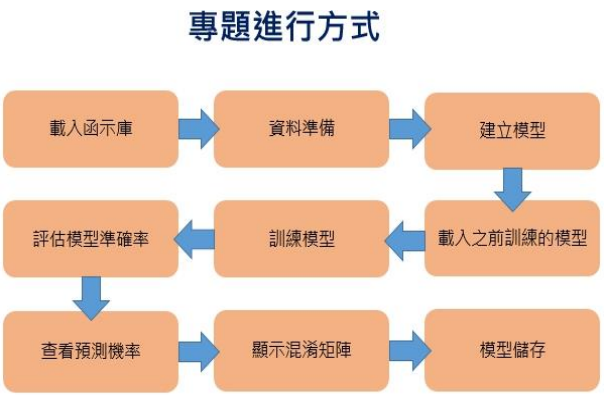
隨著資訊日漸發達，很多商業發展趨向於網路化，人人都可以上網購物，螞蟻帝國除了實體店面外也有網頁介面。螞蟻帝國創辦人希望透過螞蟻觀察，來教育民眾生態保育的知識。而在教導生態保育之前，首要步驟是先認識常見螞蟻品種。螞蟻帝國提出了合作需求，希望未來可以具有提供顧客們上傳照片即可辨識螞蟻種類的技術，因此本專題目的是與螞蟻帝國合作開發出一套影像辨識模型，來辨別台灣特有的螞蟻。

本專題參考了人工智慧實務應用的工具書[2]，來實作深度學習之螞蟻影像辨識功能，藉由多層感知器(Multilayer Perceptron)、深度神經網路 DNN(Deep Neural Network)、卷積神經網路 CNN(Convolutional Neural Network)等深度學習架構，結合螞蟻帝國提供的資料來運算模擬運作。我們剪取各品種螞蟻照片，單張照片都只有乙隻螞蟻，以供電腦辨識其特徵。

我們最後所使用的卷積神經網路中，含有多層感知器類神經網路中所沒有的卷積層及池化層。卷積層的意義是將原本一個影像，經過卷積運算產生多個影像，用來擷取照片中各個部位的特徵；再使用池化層(Max-Pool)進行縮減取樣，執行輸入的影像轉

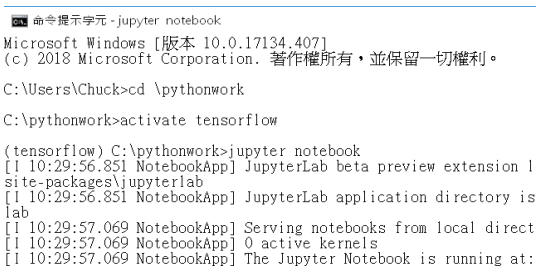
換，這樣縮減取樣會縮小影像，好處就是可以減少須處理的資料點、讓影像差異位置變小、參數的數量和計算量下降，並讓重要的特徵可以被留下[2]。

3. 專題進行方式



圖一:本專題執行步驟

本專題進行過程如圖一所示，開發環境是用 python 語言撰寫，深度學習的框架是利用 tensorflow 與 keras。硬體配備上我們採用記憶體共 16G 的筆記型電腦，因為我們在執行過程會用到大量的記憶體，這會暫存在記憶體內進行運算，如記憶體不夠大會直接無法執行；網路架構則使用深度學習的卷積式類神經網路。



圖二:啟動命令提示字元

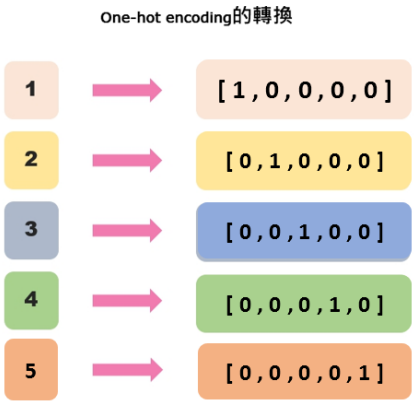
專題進行步驟的第一步，首先是在 windows 環境中啟動命令提示字元，建立 Anaconda 虛擬開發環境，接著再從 Anaconda 虛擬環境安裝 Tensorflow 與 Keras。完成後，啟動 Jupyter Notebook(如圖二)，在網頁中撰寫 python 語法，執行類神經網路的訓練。

A．載入函示庫

載入必要函示庫，如 numpy、sklearn、PIL、keras 等，其目的是為了可執行資料變換、文字處理、神經網路建立等程式。

B．資料準備

我們從螞蟻帝國所提供的螞蟻照片，對每一隻螞蟻先進行裁減，並運用套件 OpenCV 中 resize 的功能將照片設為固定大小(如 80x80)，再依照品種儲存在五個以螞蟻學名為名稱的資料夾中。程式碼中會自動取得資料夾路徑，並載入影像檔案。取得螞蟻圖檔後，再使用 one-hot encoding，將每張照片的標籤，從螞蟻分類標籤轉換成五個標註 0 或 1 的欄位(圖三)。

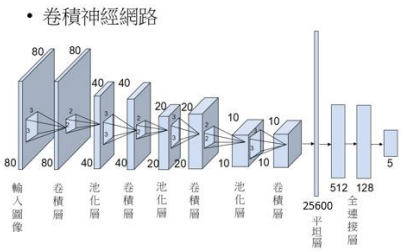


圖三:One-hot encoding 的轉換示意圖

接著，我們隨機打散照片順序，並以 8 比 2 的比例，分割訓練資料與測試資料。本專題所有螞蟻的照片總數有 778 張照片(已含水平、垂直翻轉)，其中訓練資料有 622 張，測試資料則是 156 張；然後依不同的參數設定來統一縮小照片尺寸，如 80x80 或是 160x160 來進行深度學習。

C．建立模型

如圖四所示，我們依序加入卷積神經網路的各模組，建立卷積層、池化層、神經網路之全連接層（平坦層、隱藏層）、以及最後的輸出層。可查看模型的摘要，評估網路模型的大小及參數計算量。



圖四:卷積神經網路之架構示意圖

以圖四為例，第一大層中 80x80 的照片在經過一次池化之後會變為 40x40，接著再做第二層卷積、池化變為 20x20，然後再卷積、池化變為 10x10。卷積層數可視辨識需求進行改變。每一次的卷積層會使用隨機的 filter 濾鏡，大小為 3x3，數量則有 32、64、128、256 等。濾鏡計算時，所使用的活化函數我們使用的是線性整流 ReLU。每一個濾鏡會產生一個 feature map，因此卷積之後的圖片，其厚度會增加。以圖四為例，圖中最後一次的卷積層使用了 256 個濾鏡，而照片大小為 10x10，因此最後得到的大小共為 25600 個像素值。接著進入全連接之神經網路中的第一個平坦層(flatten)節點即為 25600，再進到兩層隱藏層，隱藏層也是自由設定的參數，圖四中隱藏層的節點數分別為 512、128，這幾層之間皆是全連接。最後輸出層為五個節點，表示要對輸入的照片分類至五個分類中的哪一類。

D．載入之前訓練的模型

我們設計的程式碼中，可判斷是否為新建的模型，並開始一個新的訓練；如是之前已訓練過的模型，則會載入模型參數後，繼續訓練模型。

E．訓練模型

設定好訓練資料集與驗證資料集的比例、epoch(訓練週期)次數、每一批次筆數和是否顯示訓練過程等參數後，即可進行深度學習之模型訓練。訓練完後可以透過建立(show train history)顯示訓練過程，畫出 accuracy 執行結果，再查看是否有 overfitting(過度擬合)的現象，並畫出 loss 誤差執行結果。loss function 是使用 cross entropy error。

F．評估模型準確率

上個步驟我們建立了模型，並且完成訓練模型，接下來使用這個模型針對測試資料集中的 156 張螞蟻照片來進行螞蟻品種的偵測，評估此模型的準確率。

G．查看預測機率

使用測試資料進行預測，建立 show Predicted Probability 函數，查看每筆資料預測的機率。

H.顯示混淆矩陣(confusion matrix)

查看 predication 預測結果的形狀和 y label test 真實值的 shape 形狀再轉換為 1 維陣列後，使用 pandas crosstab 建立混淆矩陣。我們可透過此混淆矩陣來分析與探討辨識螞蟻的結果，在矩陣中，對角線上的數字表示為預測正確的資料筆數，其他非對角線上的數字則表示照片誤判的筆數。

I．模型儲存

程式最後可將訓練好的類神經網

路模型中，所有節點的權重儲存下來。此模型儲存之後，可回頭繼續執行網路的訓練。

4. 主要成果

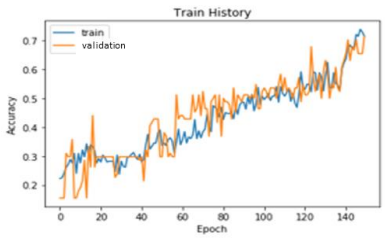
	1	2	3	4	5	6
照片大小	80X80	80X80	80X80	80X80	80X80	80X80
層數	5	5	5	5	5	5
層數組合	1,1,1,1,1	2,1,2,2,2	2,2,2,2,2	1,1,1,1,1	1,1,1,1,1	2,2,2,2,2
隱藏層	512	512,128	512	512,128	512	512
參數計算量	1,505,861	2,906,597	2,879,781	1,569,605	1,505,861	2,879,781
代數	100	100	100	100	150	150
學習準確度	0.9615	0.6218	0.5000	0.9968	1.0000	0.8619
驗證準確度	0.8000	0.4000	0.4286	0.8190	0.7738	0.5000
測試準確度	0.7619	0.4381	0.3810	0.8381	0.8381	0.5333
檔名	100-1	100-2	100-3	100-4	150-1	150-2

圖五 A: 各參數訓練結果(第 1~6 種測試)

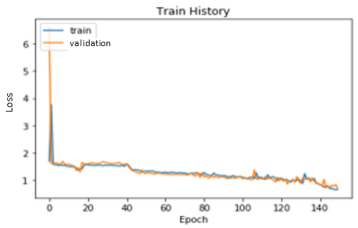
7	8	9	10	11	12	13
80X80	160X160	160X160	160X160	160X160	160X160	160x160
5	5	5	5	5	5	5
2,2,2,2,2	1,1,1,1,1,	2,2,2,2,2	1,1,1,1,1	2,2,2,2,2	2,2,2,3,3	1,1,1,1,1
512,128	128	128	128	128	128	128
2,943,525	1,798,469	3,172,389	1,798,469	3,172,389	5,696,037	1,798,469
150	100	100	150	150	150	150
0.7147	1.0000	0.9610	1.0000	0.7177	0.9960	0.9980
0.7143	0.7738	0.4881	0.7619	0.4881	0.6320	0.7440
0.7048	0.8000	0.6762	0.8476	0.6571	0.6987	0.7756
150-3	160X160(2)	100-160	160X160	160X160(1)	150(6)	150(7)

圖五 B: 各參數訓練結果(第 7~13 種測試)

如圖五，我們測試了 13 種不同的參數與層數的組合，最後挑選一個較不產生過度擬合的模型來展示(如圖六所示)。



圖六 A:訓練模型 accuracy 結果圖



圖六 B:訓練模型 loss 結果圖

一個不是過度擬合且準確率高的模型是比較理想的。在圖五的第 7 種測試組合中，我們採用五層卷積與池化，每層各有 2 次的卷積層和 1 次的池化層。在平坦層後則接著 2 層隱藏層，各 512 及 128 個節點。此模型架構最終訓練結果比較沒有過度擬合，準確率也高達七成，誤差也較小(如圖六所示)。在圖六 A 中顯示，兩條線越貼近則表示準確率結果越高，圖六 B 則表示驗證的誤差越來越低。不選擇其他準確率較高的模型，是因為那些訓練模型都有過度擬合的現象，雖然準確率高但測試準確率比學習準確率來的較低。

以這個例子來說，一開始的參數設定先用 80x80 的照片大小，將照片全部水平、垂直翻轉來增加數量；訓練模型是設定 80%作為訓練資料，20%則做為測試資料，執行 150 次的訓練，每一批次 150 筆資料然後設定顯示訓練過程，這樣才可以看到每筆資料訓練的誤差與準確率結果(如圖六)。

接下來是使用此模型來執行預測，我們可透過程式碼來看前十筆的數據以及影像的預測結果，如圖七所示。

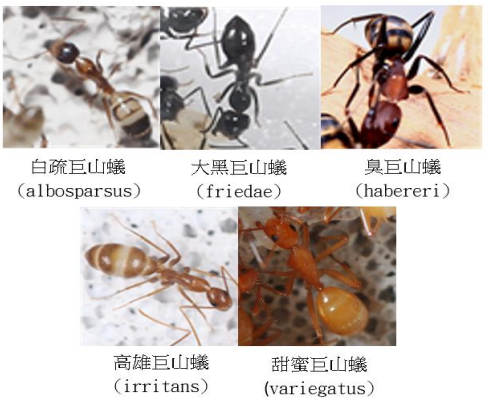
```
In [35]: prediction[:10]
Out[35]: array([2, 4, 0, 3, 3, 1, 2, 3, 0, 2], dtype=int64)
```

圖七:前十筆數據預測結果

測試資料集整體的預測結果，可以用一張混淆矩陣來表示，如圖八。測試結果顯示，白疏巨山蟻照片成功辨識的張數為 24 筆、大黑巨山蟻為 25 筆、臭巨山蟻 8 筆、高雄巨山蟻 7 筆、甜蜜巨山蟻為 28 筆；，從矩陣中可以發現，較常發生誤判的照片，如高雄巨山蟻容易被誤認為白疏巨山蟻，共 11 筆。此參數設定混淆矩陣整體的準確率為 70.48%。圖九表示本專題所研究的五種台灣特有螞蟻。

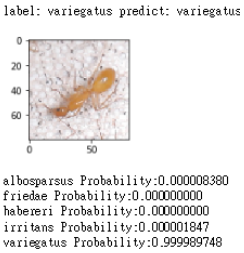
圖八:混淆矩陣圖表

我們研究發現，卷積層如果用得越少層，電腦會訓練得越好，以至訓練資料集的準確率較高，但驗證資料的學習準確率卻遠比訓練集來的差，所以訓練歷程圖中，兩資料集的曲線會分離開來，出現所謂過度擬合的現象；當我們將各卷積層數設得越多層，電腦需要計算的參數會變多，跑得時間會增加，電腦學習的深度加深之後，就較不會出現過度擬合的現象，但最終準確率則會有所下降，僅為 70%左右。本專題最終我們選用沒有過度擬合現象的結果做為本專題的成果展示(如圖六)。

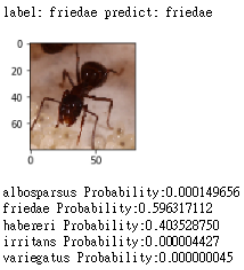


圖九:五種台灣本土特有螞蟻

由數據顯示，甜蜜巨山蟻特徵較明顯較不易混淆，故 9 張照片全部預測正確，但測試白疏巨山蟻與大黑巨山蟻、臭巨山蟻、高雄巨山蟻還是有預測錯誤的問題，以下我們挑選其中幾張辨識錯誤的照片來探討。



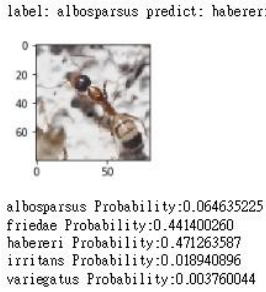
圖十 A:甜蜜巨山蟻成功辨識結果圖



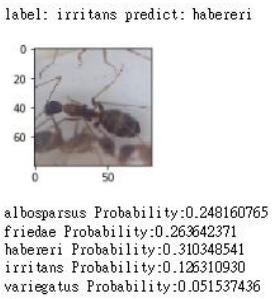
圖十 B:大黑巨山蟻成功辨識結果圖

圖十 A 與 B 表示兩項不同的預測正確結果。如圖十 A，甜蜜巨山蟻的預測正確率幾近百分之百，甜蜜巨山蟻的體色為橘黃色身形較長，特徵較為其他螞蟻明顯，因此電腦深度學習後，不會有太大的誤差或錯認的機會；而圖十 B 的大黑巨山蟻雖然也預

測正確，但從數據上可看到，大黑巨山蟻與臭巨山蟻的預測機率相近(59.63 與 40.35)，所以代表有預測錯誤的可能。我們將兩種螞蟻照片拉進比對，發現大黑巨山蟻雖然體色較黑亮，但腹部也有條紋相間，臭巨山蟻腹部則是橘色、米白色、黑色相間，所以有可能是照片明暗問題讓照片分類至多種螞蟻的機率拉近。



圖十一 A:白疏巨山蟻辨識錯誤結果圖



圖十一 B:高雄巨山蟻辨識錯誤結果圖

在辨識錯誤的螞蟻照片中，我們觀察到，即便是人眼可以輕易分辨這些螞蟻有哪些特徵是明顯不相似的，但在電腦裡，明暗度、角度、色彩、螞蟻所在的背景這些因素都會影響電腦學習時的特徵感知。例如，因為拍照光線的影響，所以這些螞蟻都看的出來是有條紋的，但色彩線條不夠明顯所以看起來就像其他種類螞蟻，如圖十一所示。

5.評估與展望

製作完這個專題後，我們期望螞蟻帝國能藉由本專題來輕鬆解決顧客對於辨識螞蟻品種的需求；此外，由於每張螞蟻的照片皆有很複雜的背景，若是在資料準備階段，能預先將螞蟻從紛亂的背景中提取出來，得到只有螞蟻本身的照片，這樣在深度網路學習後，應能提高辨識準確率。希望未來可以有人承接本專題解決照片背景問題，繼續研發出與「形色」相似的手機 APP，提供使用者可以對螞蟻拍照後，在手機上就能直接辨識是何種螞蟻。

6.結語

在這個資訊發達的世代，許多東西日漸資訊化，人人都已習慣用手機可以解決問題，翻閱書籍不再是唯一可以找到答案的方法，人們都希望能更有效的利用更多時間來做其他事，因此我們希望設計出一套適合人們操作與解決問題的系統。

7.銘謝

首先我們要感謝張慧玫老師，在一開始我們要製作 3D 列印教具時提供我們製作方向。

感謝董其樺老師對我們的耐心指導，也感謝螞蟻帝國提供我們相關資料。

8.參考文獻

[1] [機器學習專案] Kaggle 競賽-鐵 達尼號生存預測(Top 3%)
https://medium.com/@yulongtsai/https-medium-com-yulongtsai-titanic-top3-8e64741cc11f

[2] 林大貴著,”TensorFlow+Keras 深度學習人工智慧實務應用”

[3] python 出現 no module named cv2 怎麼解決？
https://www.zhihu.com/question/41519291

[4]王秉誠著，”螞蟻飼養與觀察”

[5] 螞蟻種類介紹
http://www.ant-home.idv.tw/888/a-2/a2-01.htm