

2018 Open Data 分析競賽企劃書

團隊名稱	Taiwan Deep Travel
作品名稱	深度學習為基礎的台灣旅遊景點推薦 APP
欲解決之問題與重要性	<ol style="list-style-type: none"> 1. 台灣富有優美的自然與人文場景，期望藉由旅遊推薦系統推廣台灣景點，提供貼近國內外遊客需求的旅遊資訊。此外，現代人生活忙碌，鮮少有時間搜尋旅遊資訊，倘若用戶能輸入感興趣的景點的影像至手機 APP，APP 就能推薦用戶可能喜歡的旅遊景點，將節省用戶上網查搜查資訊的時間。 2. 本 APP 包含兩種載入影像至模型的方式： <ol style="list-style-type: none"> A. 從手機裝置的圖庫（library）中上傳一張感興趣的景點影像。 B. 即時透過手機 APP 的相機功能拍攝景點影像並載入模型。 3. 本系統將即時透過深度學習影像辨識技術辨識出影像中多個場景，並依該場景推薦出與其相似的台灣旅遊景點影像，並附加該景點的詳細資訊。 4. 有別於市面上的旅遊推薦系統 APP，影像辨識為基礎的旅遊景點推薦系統的優勢與特點如下： <ol style="list-style-type: none"> A. 市面上的旅遊推薦系統缺失：大多是讓用戶以關鍵字或標籤搜尋的方式尋求偏好的景點，然而用戶卻很難以有限的文字與標籤去描述心中理想的景點，導致系統推薦的景點往往不夠精準或過於廣泛，對於用戶而言較缺乏參考價值。 B. 精準滿足用戶需求：以影像進行搜尋與推薦這種「以圖搜圖」的方式更能精準滿足用戶需求。原因在於影像能融合多項特徵，用戶不須繁瑣地輸入每項特徵於系統尋求推薦；再者，用戶往往很難用文字去形容心中理想的場景，因此以影像的表達方式將能更貼近用戶的初衷，系統所推薦的景點也能更精準符合用戶需求。 C. 著重於旅遊景點的推薦：影像辨識為基礎的旅遊景點推薦系統會著重於推薦旅遊景點，而非其他不涉及視覺欣賞的觀光體驗。然而對於富含優美自然與人文景點的台灣來說，風景欣賞本身就是台灣旅遊的一大賣點，因此該推薦系統仍具有市場價值。 D. 便捷的用戶體驗：鑒於行動裝置的普及，以 APP 的開發形式將能提供更便捷的用戶體驗，其中用戶還能隨手透過推薦系統內的相機功能拍攝周遭感興趣的景點，系統會依照照片中的場景推薦出與其相似的台灣旅遊景點影像和資訊。
開放資料之來源與使用	<ol style="list-style-type: none"> 1. 模型資料集： <ol style="list-style-type: none"> A. 來源：MIT CSAIL「Places365-Standard」。 B. 內容：Places365-Standard 資料集中提供了訓練資料集與驗證資料集，訓練資料集涵蓋 365 個場景，每個場景有 5000 張影像資料，每張影像均為 256x256x3 像素；驗證資料集也涵蓋 365 個場景，場景均與訓練資料集中的場景相同，每個場景都有 100 張影像，每張影像均為 256x256x3 像素。 C. 使用方式：從 Places365-Standard 資料集內 365 個場景中主觀

	<p>挑選出 20 個可能會出現在旅遊景點的場景。從 Places365-Standard 資料集內的訓練資料集中取得這 20 個場景，每個場景有 5000 張影像資料，並將這 20x5000 筆影像資料做為模型的訓練資料；從 Places365 資料集內的驗證資料集中同樣地取得這 20 個場景，每個場景有 100 張影像資料，並將這 20x100 筆影像資料做為模型的驗證資料。最後將這 20x5000 筆訓練資料與 20x100 筆驗證資料輸入卷積神經網路 (Convolutional Neural Network; CNN) 做模型訓練與驗證。</p> <p>D. 挑選 20 個場景來訓練模型的理由：</p> <p>I. 為了能在 APP 上順利運作模型，模型勢必不能太大。如果可以只挑選 20 個場景來訓練則可以降低模型的總參數量與大小。</p> <p>II. 訓練能辨識越多場景的模型可能導致模型辨識準確率下降。</p> <p>III. 完整訓練能辨識 365 個場景的模型所需時間較久。所以僅挑選出 365 個場景中的 20 個作為訓練。</p> <p>2. 所推薦的景點資料集：</p> <p>A. 來源：政府資料開放平台中由交通部觀光局所提供的「景點 - 觀光資訊資料庫」。</p> <p>B. 內容：資料集涵蓋 4809 個景點，地點遍及全台，每個景點有 1 張影像資料與關於景點的文字介紹。</p> <p>C. 用途：當用戶隨機輸入一張景點的影像資料至本 APP 中，系統將產生一個 1x20 的向量，向量中的每個值分別代表模型預測各場景的機率。同時，所推薦的景點資料集中的每張影像資料也將輸入至影像辨識模型中並產生 4809 個 1x20 的向量。用戶輸入的影像所產生的向量與 4809 個觀光景點所產生的向量兩者進行 4809 次的餘弦相似度比較，最後將相似度前五名的觀光景點推薦給用戶。</p> <p>MIT CSAIL 實驗室開源「Places365-Standard 資料集」： http://places2.csail.mit.edu/download.html</p> <p>政府資料開放平台「景點 - 觀光資訊資料庫」： https://data.gov.tw/dataset/7777</p>
<p>人工智慧與資料科學技術使用狀況之說明</p>	<p>1. Keras：Keras 是高階的深度學習框架，以 Python 程式語言撰寫，能夠運行在 Tensorflow 之上。具有簡易和快速設計的特性，同時也支持卷積神經網路和 GPU 運算。本團隊使用 Keras 深度學習框架進行卷積神經網路的實作與訓練。</p> <p>2. Google 雲端運算平台：使用 Google 雲端運算平台 (Google Cloud Platform; GCP) 所提供的計算引擎 (compute engine) 服務建立虛擬機器 (virtual machine) 執行個體，虛擬機器的規格設定如下：</p> <p>A. 兩顆 GPU，13GB 記憶體。</p> <p>B. 作業系統：Windows Server 2016。標準永久磁碟：200GB。</p> <p>C. GPU 規格：NVIDIA Tesla K80，24GB 記憶體。</p> <p>再以 Google 雲端運算平台所提供的 RDP (Remote Desktop Protocol) 檔進行遠端操控。並且在虛擬機器上安裝 GPU 驅動程式與 Python 套件。</p>

3. 資料處理：

- A. 將訓練資料集 20 個場景的所有圖片路徑打散，使路徑隨機排序。
- B. 將訓練及驗證資料集 20 個場景的圖片 256x256x3 像素轉換成 224x224x3 像素，以符合後續透過遷移學習（transfer learning）所採用的 ResNet50 模型輸入大小。
- C. 將訓練及驗證資料集的影像值標準化，可以提高後續訓練模型預測的準確度，並使梯度運算時收斂的更快。
- D. 透過 Keras Image Data Augmentation 來擴增驗證資料集，將原本 20x100 張影像資料擴增（旋轉、調整大小、比例尺寸，或者改變亮度色溫、翻轉等處理），再將擴增後的驗證資料集隨機分成驗證資料集與測試資料集，驗證資料集用於決定模型的超參數，測試資料集用於評估最終模型。

4. 建立模型：

- A. 採用遷移學習的方式。僅修改模型架構的全連接層（fully connected layer），原因在於先前嘗試過重新訓練 VGG16 模型，然而訓練 90 週期之後驗證準確率僅 52%，如果再考慮調整超參數所需要的時間，就整體成效而言並不具效率。

重新訓練的 VGG16 模型 90 週期結果程式碼：

https://drive.google.com/open?id=1LtAsvnm84vnziVRKpVXA1ex6TB_vyLN

- B. 選擇遷移學習的模型為 ResNet50。ResNet 為 Kaiming He 於 2016 年所獲得的 CVPR 最佳論文，且在 ILSVRC 2015 比賽中獲得了冠軍，其加深了神經網路同時也解決了深度網路所出現的退化（degradation）問題。Keras 應用模塊（keras.applications）提供各種影像辨識模型預訓練模型（pre-trained model）的儲存權重的佔據量、top-5 正確率、top-1 正確率。由數據可知，ResNet50 的 top-1 正確率和 top-5 正確率均優於 VGG16，而 ResNet50 儲存權重的佔據僅為 99MB 明顯小於 VGG16 的 528MB，有利於鑲嵌在行動裝置上。所以本團隊所選擇實作遷移學習的模型為 ResNet50。
- C. 目前實際測試後，在尚未調整超參數的情況下，遷移學習的 ResNet50 模型在訓練 19 週期之後訓練資料的準確率為 97%，測試資料的準確率為 71%；同時比較遷移學習的 ResNet50 和遷移學習的 VGG16 模型在訓練 5 週期之後訓練資料的準確率分別為 86%和 58%；測試資料的準確率分別為 68%和 57%。

利用遷移學習訓練 ResNet50 模型共 19 週期訓練結果程式碼：

https://drive.google.com/open?id=13AeJtwtrNwr05IpQiqPxz0U-5k6_LfAg

同時比較利用遷移學習訓練 ResNet50 模型和遷移學習訓練 VGG16 模型結果的程式碼：

<https://drive.google.com/open?id=1VdjXL6ujiRd0iHoT15JcX2-Ky8QIFd9T>

- D. 預計新增 Batch Normalization 層於激活函數層之上。Batch Normalization 的概念是 Google 於 2015 年的論文中所提出，目的是為了防止梯度擴散的問題並確保每一層中的輸入資料分佈是穩定的，進而達到加速訓練的目的。
- E. 本團隊目前已嘗試用 ReLU 作為激活函數；Adam 作為優化演算法

	<p>建模，訓練與驗證資料準確率分別為 97%與 71%，本團隊預計將修改全連接層中之激活函數並改採用 SeLU；SeLU 在 arXiv 上公開的一篇 NIPS 投稿論文《Self-Normalizing Neural Networks》中被提出，能夠保證梯度在反向傳遞的過程中不會出現梯度消失或者是梯度爆炸的問題，同時保證激活輸出能夠是標準的高斯分佈，在 MNIST 上測試後發現效果確實比 ReLU 好，收斂速度快。</p> <p>F. 本團隊預計優化演算法將採用 Adafactor。Adafactor 為 Google 於 2018 年基於 Tensorflow 新開源的深度學習庫 Tensor2Tensor 中的優化演算法，針對目前主流優化方法如 SGD、Adam 所存在的問題進行改進，其中在記憶體消耗、參數更新以及遞迴步長三個方面對 Adam 算法進行了改進。</p> <p>5. 模型訓練：隨著訓練週期增加，僅儲存比上一個週期更好的驗證損失函數(loss function)值的權重。</p> <p>6. 模型驗證：調整超參數(hyperparameters)，以處理過度配適(overfitting)的問題。其中分別調整學習速率(learning rate)、dropout 和激活函數(activation functions)。</p> <p>7. 模型測試：輸入由擴增後的驗證資料集所分割出的測試資料於模型進行測試以評估最終模型。</p> <p>8. 推薦系統：</p> <ul style="list-style-type: none"> A. 輸入 4809 個台灣觀光景點影像資料於模型，每筆影像資料都將產生對應於 20 種場景的預測機率，並各自以 1x20 的向量儲存。 B. 當用戶輸入一張手機圖庫中的圖片或隨手拍攝的照片進入影像辨識模型中，模型同樣會產生透過影像辨識所預測的 20 種典型旅遊場景的機率，並以 1x20 的向量儲存。 C. 將用戶輸入的一個影像資料的 1x20 的機率向量和台灣觀光景點影像資料 4809 個 1x20 的機率向量做餘弦相似度比較，系統將推薦用戶餘弦相似度的數值前五高的台灣景點影像和資訊。 <p>9. 用戶介面：</p> <ul style="list-style-type: none"> A. 將本系統所有的 Python 程式碼人為轉換成 Swift 程式語言程式碼，並透過蘋果公司的 Xcode 整合開發環境將系統開發成 iOS 手機 App。 B. 透過蘋果公司的 Core ML 框架將 Keras 模型的 .h5 格式檔轉換成 iOS 作業系統可以讀取的 .mlmodel 格式檔，並將模型載入 Xcode 開發環境。Core ML 2.0 框架為蘋果公司於 2018 在 WWDC 大會上所發表的開發機器學習框架，其能快速將已完成訓練的機器學習模型整合進 APP 中。
<p>預期完成之功能或結果</p>	<p>1. 目前已完成之結果：</p> <ul style="list-style-type: none"> A. 從 Places365 資料集內 365 個場景中主觀挑選出 20 個會出現在旅遊景點的場景。場景部份圖片(每個場景各 10 張)與場景清單： https://drive.google.com/drive/folders/1nPDub31S2NG4rcvLbwhjHULZOC-Qrml7?usp=sharing B. 20 個場景訓練及驗證資料預處理程式碼： https://drive.google.com/open?id=1laAprk3EBhHSdNhc-a51ETuzVaxuxRSE

	<p>C. 利用遷移學習訓練 ResNet50，僅修改全連接層，激活函數採用 ReLU；優化演算法採用 Adam；Dropout 設定為 0.5；訓練 19 週期結果訓練與驗證準確率分別為 97%與 71%，程式碼連結： https://drive.google.com/drive/u/1/folders/13AeJtwtrNwr05IpQiqPxz0U-5k6_LfAg</p> <p>D. 隨機挑選 100 個台灣景點，每個景點的影像各 10 張共 1000 張影像進行 APP 系統測試。將 100 x 10 張影像資料輸入模型，同時模擬用戶情境隨機輸入任意一張影像進系統，兩者進行相似度比較。最後系統列印出前五名相似的台灣景點。程式碼： https://drive.google.com/open?id=1kDZjAc2ZDa2PyqoQyxMteWq2DCgvys6</p> <p>E. 完成推薦系統 iOS APP 原型與實機測試，DEMO 影片連結： https://drive.google.com/open?id=1gwS8ikDXwAoUBFxpSL-KIGGXgGE3Pg0x</p> <p>2. 預期完成的結果：</p> <p>A. 透過 Keras Image Data Augmentation 來擴增驗證資料集，將原本 20x100 張影像資料擴增，再將擴增後的驗證資料集隨機分成驗證資料集與測試資料集，驗證資料集用於決定模型的超參數，測試資料集用於評估最終模型。</p> <p>B. 新增 Batch Normalization 層於激活函數層之上。</p> <p>C. 為使 APP 能更方便地應用於行動裝置，將考慮一些近年來提出更輕量化之卷積神經網路，並評估其表現（例如：MobileNet、DenseNet、NASNetMobile 等）。</p> <p>D. 使用 SeLU 和 Adafactor 作為模型的激活函數與優化演算法。</p> <p>E. 在驗證模型時執行調整超參數的動作來處理過度配適的問題。</p> <p>F. 輸入由 Keras Image Data Augmentation 擴增後的驗證資料集分割出的測試資料於模型進行測試。</p> <p>G. 比較餘弦相似度和改良的餘弦相似度做景點推薦的效果。</p> <p>H. 優化 APP 的用戶介面，除了呈現推薦的景點影像外，還附加關於該景點的詳細資訊。</p> <p>I. 蒐集用戶評估建議。</p> <p>3. 文獻參考：本團隊參考之相關文獻如下 https://drive.google.com/open?id=1iNjNm1ACJjJQNwqBT93agqv1qtub0Dh3</p>
其他補充說明	<p>克服瓶頸：</p> <p>A. CPU 運算速度過慢問題：由於卷積神經網路模型的訓練資料通常都非常龐大，經過測試，CPU 與 GPU 的運算速度相差約 30 倍。本團隊花了許多時間成功研究出在遠端連線 Google Cloud Platform 下使用虛擬機器運行 GPU，才解決運算速度過慢的問題。</p> <p>B. Keras 模型轉行動裝置模型問題：在使用蘋果（Apple）公司的 Core ML 框架將 Keras 模型的 .h5 格式檔轉成 iOS 作業系統可以讀取的.mlmodel 格式檔時，產生 ReLU6 參數匯入錯誤。最後與 Stackoverflow 網站的人討論之後才發現是 Keras 2.2.2 版本之後的套件中沒有 ReLU6 這個參數，但 coremltools 套件持續存在匯入 ReLU6 這個參數的指令。將 coremltools 套件中的程式碼進行修改問題才得以解決。</p>