2018 Open Data 分析競賽成果說明書

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
報告人: 吳政翰，李奇
指導教授: 張揖平

一、 欲解決之問題與重要性

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。本 APP 包含兩種載入影像至模型的方式: 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。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。

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：

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。

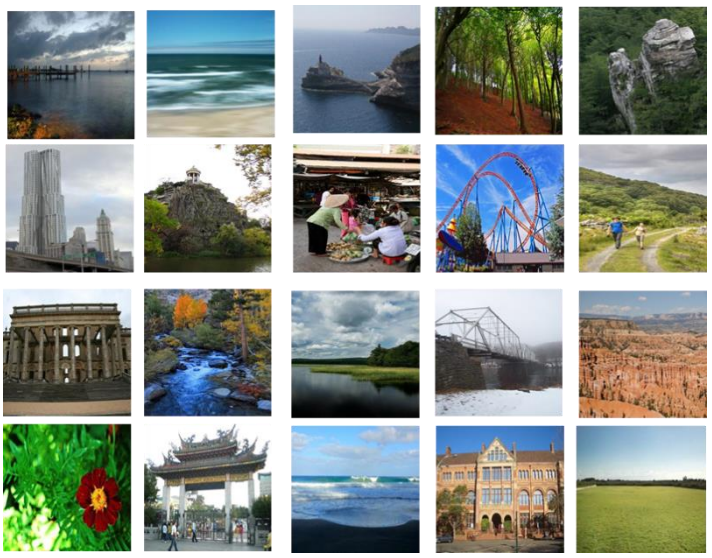
C. 著重於旅遊景點的推薦：影像辨識為基礎的旅遊景點推薦系統會著重於推薦旅遊景點，而非其他不涉及視覺欣賞的觀光體驗。然而對於富含優美自然與人文景點的台灣來說，風景欣賞本身就是台灣旅遊的一大賣點，因此該推薦系統仍具有市場價值。

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，系統會依照片中的場景推薦出與其相似的台灣旅遊景點影像和資訊。

二、 開放資料之來源與使用

2.1 模型資料集

來源為 MIT CSAIL「Places365-Standard」[1]。內容包含 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。使用方式是從 Places365-Standard 資料集內 365 個場景中主觀挑選出 20 個可能會出現在旅遊景點的場景。



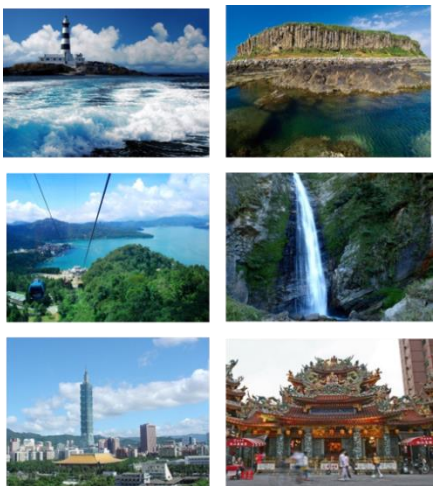
從 Places365-Standard 資料集內的訓練資料集中取得這 20 個場景，每個場景有 5000 張影像資料，並將這 20x5000 筆影像資料做為模型的訓練資料；從 Places365 資料集內的驗證資料集中同樣地取得這 20 個場景，每個場景有 100 張影像資料，並將這 20x100 筆影像資料做為模型的驗證資料。最後將這 20x5000 筆訓練資料與 20x100 筆驗證資料輸入卷積神經網路 (Convolutional

[1] MIT CSAIL 實驗室開源「Places365-Standard 資料集」：http://places2.csail.mit.edu/download.html

Neural Network；CNN) 做模型訓練與驗證。挑選 20 個場景來訓練模型的理由是為了能在 APP 上順利運作模型，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，以防模型太大。再者，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。最後是時間上的考量，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。

2.2 所推薦的景點資料集

來源是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」[2]。內容包含資料集涵蓋 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。用途為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。



三、 技術說明

3.1 使用工具

應用 Keras 作為深度學習工具。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

[2]政府資料開放平台「景點 - 觀光資訊資料庫」： https://data.gov.tw/dataset/7777

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。另外也使用 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，虛擬機器的規格包含兩顆 GPU，13GB 記憶體。作業系統是 Windows Server 2016。標準永久磁碟為 200GB。 GPU 規格為 NVIDIA Tesla K80，24GB 記憶體。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。

3.2 資料處理

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。接著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。另外還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。最後透過 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。

3.3 建模流程

以下建模皆是訓練所有層 5 週期，並且驗證資料為 2000 張擴充後驗證資料。

3.3.1 決定合適的遷移學習模型 (Transfer Learning)

遷移學習模型用神經網路的詞語來表述，就是一層層網路中每個節點的權重從一個訓練好的網路遷移到一個全新的網路里，然而並不需要重新訓練每層網路的權重，而是僅修改模型架構的全連接層 (fully connected layer)，原因在於從頭建立模型復雜且耗時，通過遷移學習將可以加快學習效率。

分別比較 VGG16、ResNet50、MobileNet、DenseNet、NasNetMobile 與 MobileNetV2 六種遷移學習模型的效能。VGGNet 為牛津大學計算機視覺組（Visual Geometry Group）和 Google DeepMind 公司的研究員於 2014 年一起研發的的深度卷積神經網絡。同時 VGGNet 的拓展性很強，遷移到其他圖片數據上的泛化性非常好。VGGNet 成功地構築了 16~19 層深的卷積神經網絡。ResNet 為何凱明 於 2016 年所獲得的 CVPR(IEEE 国际计算机视觉与模式识别会议) 最佳論文，且在 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2015 比賽中獲得了冠軍。解決了隨著網路層數加深所產生的梯度消失(vanishing gradient)與梯度爆炸(gradient exploration)。相對於 VGGNet，ResNet 成功地構築了 50、101、152 層更深度的神經網路。MobileNet:
為 Google 於 2017 提出，屬於輕量的神經網路模型，擁有較低的模型全權重與參數的儲存空間，解決過度複雜的模型無法在手機等低階產品運行的問題。Google 團隊也使用 MobileNet 在不同的影像識別任務均有不錯的效果。DenseNet 則榮獲 CVPR 2017 的最佳論文獎，相對於 ResNet 提升了梯度的反向傳播，使得模型更容易訓練。由 DenseNet 在 CIFAR-100 和 ImageNet 數據集上與 ResNet 的對比結果來看，DenseNet 不論是預測誤差或參數大小都小於 ResNet。

表一為利用遷移學習 (Transfer Learning) 訓練各模型，訓練所有層 5 週期後的結果。可以發現 MobileNet 不論在準確度、大小與花費時間上都有較佳的效果。

表一

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
VGG16 (2014 ILSVRC)	0.8538	0.7390	0.9250	0.9885	0.6760	0.9160	266 MB	23,238,356	27	47 ms/step
ResNet50 (2015 ILSVRC)	0.3632	0.8760	0.9800	1.0926	0.7060	0.9140	417 MB	36,435,476	180	40 ms/step
MobileNet (2017 Google)	0.9317	0.7000	0.9170	1.0584	0.6940	0.9030	110 MB	9,654,100	92	17 ms/step
DenseNet121 (2017 CVPR)	0.7228	0.7650	0.9350	0.9701	0.7030	0.9150	154 MB	13,462,740	431	38 ms/step
NASNetMobile (2018)	0.7656	0.7650	0.9320	0.8591	0.7320	0.9250	128 MB	10,895,656	774	34 ms/step
MobileNetV2 (2018 Google)	0.8226	0.7480	0.9240	0.9828	0.6780	0.9120	118 MB	10,288,852	160	20 ms/step

3.3.2 決定最優化算法

比較 Adam 與 Adafactor 兩者最優化算法用在 MobileNet 何者為佳。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、參數更新以及學習率(learning rate)三個方面對 Adam 算法進行了改進。表二為運用 Adam 與 Adafactor 作為最佳花算法，訓練所有層 5 週期後，並且驗證資料為 2000 張擴充後驗證資料的效能比較。可以明顯看出用 Adam 作為最優化算法的效能較佳。

表二

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
MobileNet with Adam (2015)	0.9624	0.6760	0.9030	1.5239	0.5565	0.8020	110 MB	9,654,100	92	22 ms/step
MobileNet with Adafactor (2018)	3.2579	0.0510	0.1300	3.4619	**0.0500**	0.1500	37 MB	9,654,100	92	21 ms/step

3.3.3 決定 Batch Normalization (BN) 與激活函數

Google 於 2015 年的論文中所提出。透過增設 Batch Normalization 於神經網絡中的每層全連接層與激活函數間，將能使在每一層中輸入到激活函數的數據的分布更穩定，以防止梯度消失 (vanishing gradient) 的問題。表三比較了 ReLU with BN 、 ReLU without BN 與 SeLU with BN 三種組合。其中 ReLU without BN 的組合最佳。

表三

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
ReLU with BN	0.8410	0.7250	0.9230	1.3066	**0.5835**	0.8315	117.7 MB	9,855,316	95	21 ms/step
ReLU without BN	0.9624	0.6760	0.9030	1.5239	**0.5565**	0.8020	110 MB	9,654,100	92	22 ms/step
SeLU with BN (2015)	0.7213	0.7840	0.9390	1.3839	**0.5695**	0.8215	117.7 MB	9,855,316	94	20 ms/step

3.3.4 決定動態或靜態 Learning Rate

由表四可知 learning rate 為靜態且在 0.0002 時有最佳的結果。

表四

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
MobileNet (LR = 0.00002)	0.8410	0.7250	0.9230	1.3066	0.5835	0.8315	117.7 MB	9,855,316	95	21 ms/step
MobileNet (LR = 0.0002)	0.7651	0.7460	0.9360	1.3139	**0.5800**	0.8400	112 MB	9,855,316	95	21 ms/step
MobileNet (LR = 0.002)	1.1672	0.6360	0.8590	1.7032	0.4620	0.7455	112 MB	9,855,316	95	22 ms/step
MobileNet (LR = 0.002 _ 0.00002)	1.0588	0.6670	0.8770	1.4206	0.5540	0.7965	112 MB	9,855,316	95	21 ms/step

3.3.5 調整超參數

L1 正規化和 L2 正規化可以看做是損失函數的懲罰項，期能對損失函數中的某些參數做一些限制來降低模型複雜度，以解決 overfitting 的問題。

表五

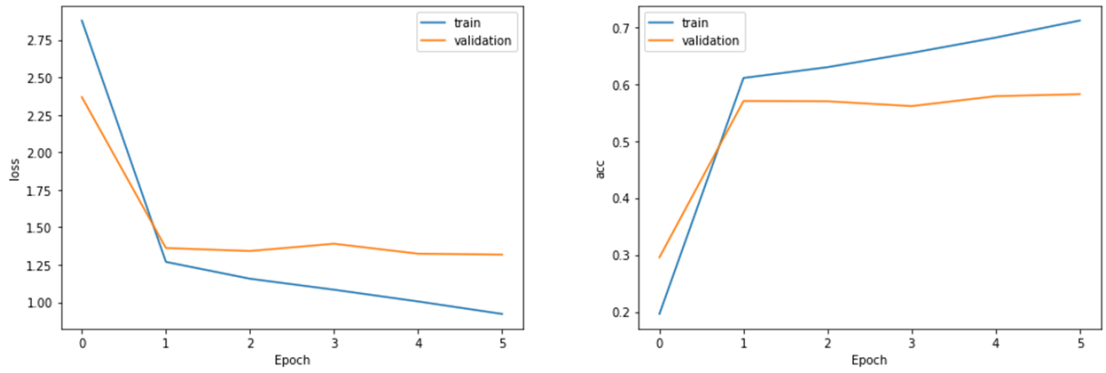
kernel_regularizer activity_regularizer	Dropout	Dense	Loss	Top-1 Acc	Val loss	Top-1 Val_acc	Size	Parameters	Depth	Time
0	0.5	128	0.9438	0.6880	1.2661	0.5840	112 MB	9,855,316	94	21 ms/step
0	0.5	256	0.8881	0.6970	1.2515	0.5910	185 MB	16,281,044	94	22 ms/step
0	0.2	128	0.8632	0.7090	1.2756	0.5700	112 MB	9,855,316	94	21 ms/step
0	0.2	256	0.9959	0.6630	1.2755	0.5780	185 MB	16,281,044	94	21 ms/step
0.01	0.5	128	4.4488	0.0500	4.4374	0.0510	117.7 MB	9,855,316	94	22 ms/step
0.01	0.5	256	5.6078	0.1880	5.3630	0.2795	194.8 MB	16,281,044	94	21 ms/step
0.01	0.2	128	3.6754	0.3710	3.6977	0.3865	117.7 MB	9,855,316	94	22 ms/step
0.01	0.2	256	5.4007	0.4580	5.4992	0.4165	194.8 MB	16,281,044	94	22 ms/step

3.4 模型表現評估

使用 3.3 建模流程中每個環節所選出的最佳模型與參數，並應用於訓練與驗證資料進行訓練，最後再運用測試資料集進行模型測試。分別訓練 5 週期與 10 週期進行比較

3.4.1 訓練 5 週期的模型訓練與測試結果

訓練5週期下，訓練資料top1正確率為0.69，驗證資料top1正確率為0.5835 (表六)。測試資料 top1 正確率為 0.614 (表七)。



圖一

表六

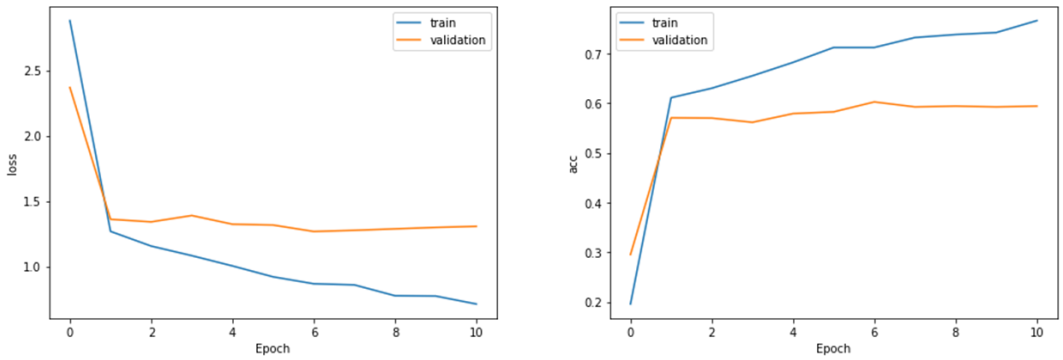
Model	Loss	Top-1 Acc	Top-3 Acc	Val_loss	Top-1 Val_acc	Top-3 Val_acc
5週期	0.9600	0.6900	0.9020	1.2687	**0.5835**	0.8515

表七

Model	Loss	Top-1 Acc	Top-3 Acc
5週期	1.2499	**0.5970**	0.8510

3.4.2 訓練 10 週期的模型訓練與測試結果

訓練 10 週期下，訓練資料 top1 正確率為 0.738，驗證資料 top1 正確率為 0.607 (表八)。測試資料 top1 正確率為 0.614 (表七)。



圖二

表八

Model	Loss	Top-1 Acc	Top-3 Acc	Val_loss	Top-1 Val_acc	Top-3 Val_acc
5_5週期	0.8040	0.7380	0.9360	1.2467	**0.6075**	0.8535

表九

Model	Loss	Top-1 Acc	Top-3 Acc
5_5週期	1.2206	0.6140	0.8465

3.5 推薦系統

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。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。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。

餘弦相似性通過測量兩個向量的夾角的餘弦值來測量它們之間的相似性。當兩個向量相同時，餘弦相似度的值為 1；兩個向量無關時，餘弦相似度的值為 0；兩個向量完全負相關時，餘弦相似度的值為-1。餘弦相似度更多的是從方向上區分差異。然而它沒辦法衡量每個維數值的差異，對絕對數值的不敏感導致了結果

的誤差，因此需要修正後的餘弦相似度 (圖三) 來解決著個問題。修正後的餘弦相似度在計算時，會將向量中的每個維度的元素減去該維度對應的均值作為計算值。

$$\frac{\sum_{i=1}^{20}[(A_i - \boldsymbol{\mu_A}) \times (B_i - \boldsymbol{\mu_B})]}{\sqrt{\sum_{i=1}^{20}(A_i - \boldsymbol{\mu_A})^2}\sqrt{\sum_{i=1}^{20}(B_i - \boldsymbol{\mu_B})^2}}$$

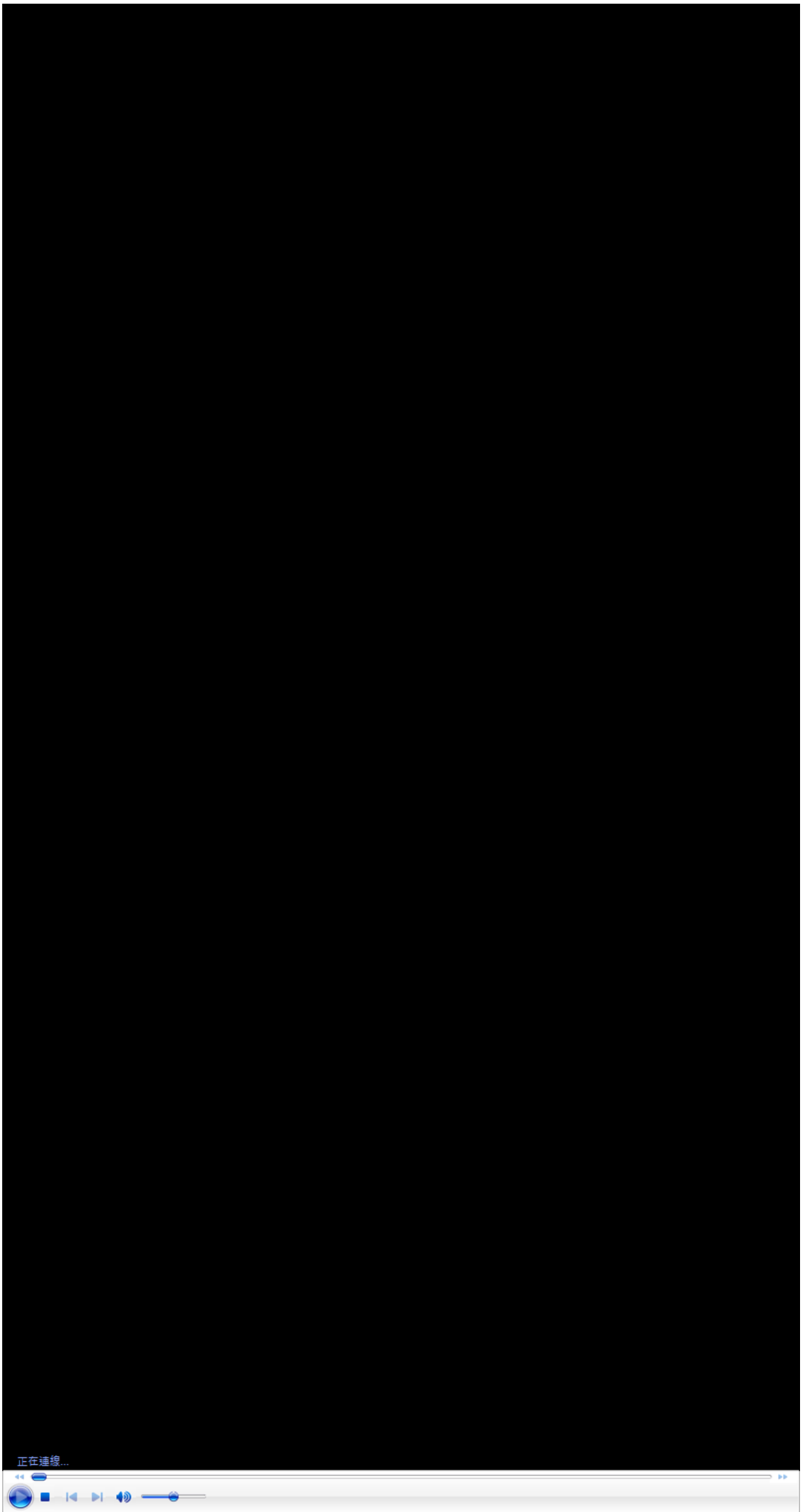
圖三

3.6 使用者介面

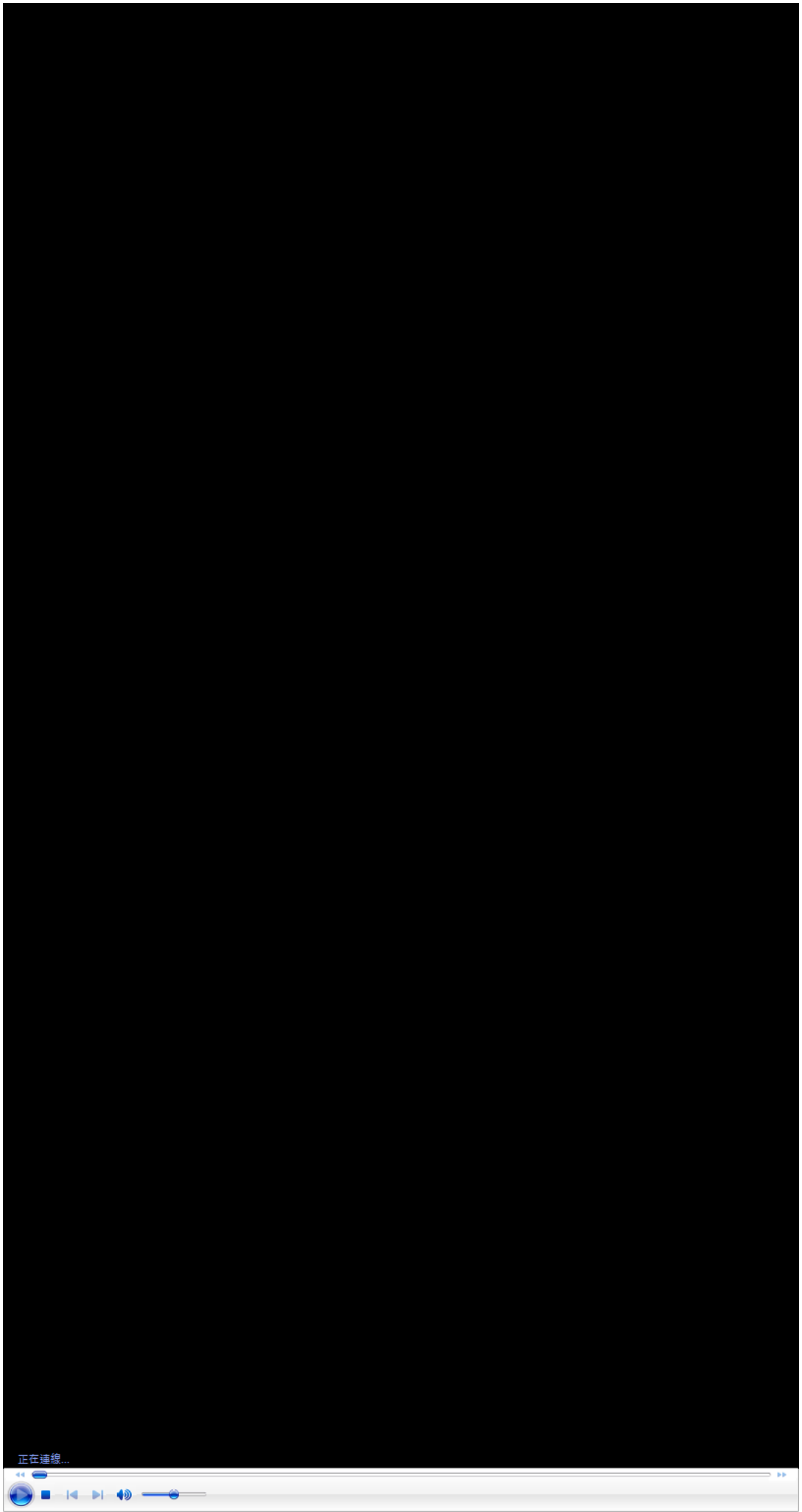
將本系統所有的 Python 程式碼人為轉換成 Swift 程式語言程式碼，並透過蘋果公司的 Xcode 整合開發環境將系統開發成 iOS 手機 App。透過蘋果公司的 Core ML 框架將 Keras 模型的 .h5 格式檔轉換成 iOS 作業系統可以讀取的 .mlmodel 格式檔，並將模型載入 Xcode 開發環境。Core ML 2.0 框架為蘋果公司於 2018 在 WWDC 大會上所發表的開發機器學習框架，其能快速將已完成訓練的機器學習模型整合進 APP 中。

四、 實機展示

4.1 從手機圖庫中載入圖片


正在連線...

4.2 以手機即時拍攝圖片並載入



五、 研究展望

為了增加模型辨識場景的能力與推薦景點的精準度，期望以更多場景訓練模型以提升模型。另外在模型準確率方面，為了提升模型準確率，期望擴增訓練樣本供模型訓練以提升驗證與測試資料的準確率，因為在本研究中只有驗證與測試資料包含了被擴增的影像資料，訓練資料集中並沒有被擴增的影像資料，然而擴增的影像資料與原始影像資料的型態本身仍存在一定落差，如果沒有訓練模型跑過這些資料，最終以現有的模型架構去驗證與測試被擴增的資料時準確率將下降。而在調整超參數方面，可以加入更多超參數組合進行比較。在推薦景點的部分，期望可以加入更多 Open Data 以豐富系統推薦的景點。另外也可以利用 TensorBoard 來視覺化神經網絡。最後再 APP 方面，可以將系統擴展到 Android 手機作業系統上，並蒐集用戶對 APP 的評估建議。

六、 其他補充

在本次研究中克服了 CPU 運算速度過慢與 Keras 模型轉行動裝置模型等問題。CPU 運算速度過慢方面，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。Keras 模型轉行動裝置模型問題方面，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。

七、 參考文獻

- 洪文麟。2016。深度學習應用於以影像辨識為基礎的個人化推薦系統-以服飾樣式為例。碩士論文。台南: 成功大學工程科學研究所

- Audrey Tam. 2018. Beginning Machine Learning with Keras & Core ML. http://www.raywenderlich.com/188-beginning-machine-learning-with-keras-core-ml
- Justin.h. 2017.图片数据集太少？看我七十二变，Keras Image Data Augmentation 各参数详解. github.com/JustinhoCHN/keras-image-data-augmentation
- Keras Documentation. Available at keras.io/applications/
- Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 2015. Deep Residual Learning for Image Recognition
- MIT CSAIL Com. 2018. Release of Places365-CNNs. github.com/CSAILVision/places365n
- Shao-Hua Sun. 2017. SELUs (scaled exponential linear units) - Visualized and Histogramed Comparisons among ReLU and Leaky ReLU. github.com/shaohua0116/Activation-Visualization-Histogram
- Stackoverflow. 2018. Cosine Similarity between 2 Number Lists. stackoverflow.com/questions/18424228/cosine-similarity-between-2-number-lists