

AI-capstone Project 1

姓名：洪翌洋

學號：112550170

Code : <https://github.com/yyh-603/google-street-view>

Dataset : <https://github.com/yyh-603/google-street-view-dataset>

Dataset 與 112550006 周宜勳 共同製作

Introduction

啟發自 Geoguessr 這款遊戲，本次作業主題為台灣各縣市街景分類，透過 Google map API 蒐集台灣街景圖片，並進行資料清洗，最終使用 K-means、SVM、ResNet 進行訓練，並比較多種模型之間的差異。訓練街景分類的模型，可以應用於相片標籤，地理特徵分析等問題上。

Dataset

簡介

本次 dataset 主題為台灣本島街景圖片，資料來源為 Google 街景服務。我們使用 Google map API 進行圖片蒐集，蒐集的資料包含經度、緯度、縣市名稱、地址，並用 csv file 記錄圖片名稱與其對應的資訊。

dataset 中包含台灣本島 19 個縣市，每個縣市隨機位置 300 張街景照片、照片朝向隨機、仰角固定為 0、可視範圍固定 90，圖片大小固定 640x640，並且去除室內及私有地照片。



蒐集方式

Google map 提供許多 API，以下列出本次作業使用到的 API，以及在蒐集程式中的角色。

1. Google Street View Static API：此 API 可以獲得街景圖片，需要指定經緯度、朝向、仰角、可視範圍等。

2. Google Street View Metadata：此 API 可以確認指定的經緯度有沒有街景照片，在確認該地點街景服務後，程式才會使用 Static API 向 Google 請求圖片，減少資源以及成本。

3. Google Geocoding：此 API 可以指定經緯度，並獲得該位置的地址。程式使用此 API 獲得縣市與地址資訊。

實作

我們先手動將每個縣市的大致經緯度範圍建表，程式遍歷每個縣市，在每個縣市的經緯度範圍內隨機選位置，檢查該位置是否有街景服務，若有服務及檢查該地點是否為目標城市。確認以上資訊後，程式才會向伺服器要求街景照片，並且寫入 csv。

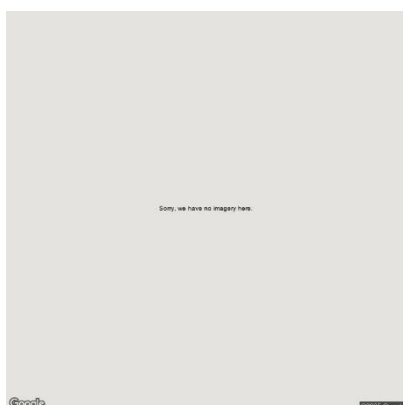
資料清洗

雖然以上爬蟲盡可能減少獲得無用資料的可能性，但實際上還是有部分資料需要重新處理。

1. 未找到圖片：在爬蟲的過程中，可能因為一些不明原因，google 找不到對應的圖片，此時會回傳預設照片，需要將這類照片去除。(如下左圖)

2. 室內照片：因為 API 無法確認是否是室內的相片，在爬蟲過程中無法處理。有部分明顯是非公共區域的相片(如下右圖)，這部分照片也會一併去除。

上述兩種照片我們會手動去除，並且另外補上這些資料。



Algorithm

1. Principal components analysis (PCA)：本次實驗主要輸入為圖片，若是直接輸入圖片，可能因為圖片維度過高，導致模型訓練時間過長，並且存在 Curse of dimensionality 的問題。PCA 透過將原始向量投影到較低維度空間，並盡可能增加變異量，以此在保留原始數據的特徵下降低維度。

2. Support Vector Machine (SVM)：SVM 是一種 supervised learning，目標是在資料中找到一個 Hyperplane，將兩群資料切開，並且讓類別距離此 Hyperplane 最大化。這份作業中使用的 kernel function 為 Radial basis function，可以做到非線性的分類。

3. K-means：K-means 透過隨機選取中心點，並且根據距離移動中心點，目標讓數據距離中心點最小，達到 unsupervised 的效果。本次作業中資料有 label，因此在做完 K-means 後，會選擇該群最多的標籤作為該群的分類。

4. Residual Network (ResNet)：ResNet 是一種深度學習架構，利用 Residual Block 以及 Shortcut Connection 解決梯度消失以及梯度爆炸的問題，讓 ResNet 能夠在層數增加的情況下能夠順利進行訓練。本次作業會使用 pytorch 的 pretrained model。

Experiments & Analysis

K-means

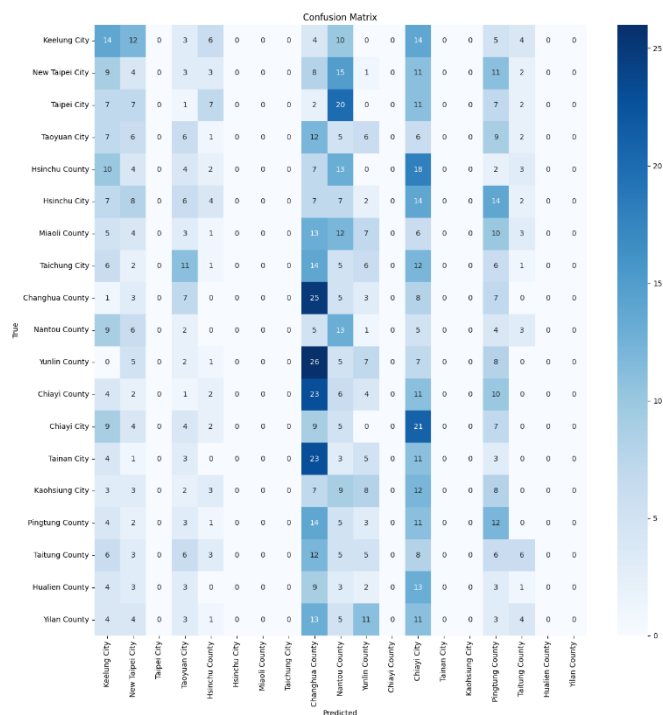
使用 K-means 進行實驗。實驗會先使用 PCA 將圖片降維，再使用 K-means 進行分群以及預測。主要調整參數有 K-means 的 n-cluster 以及 PCA 的 feature 數量。

1. 實驗結果：

(1) accuracy：



(2) confusion matrix (取 accuracy 最高者展示)：



2. 分析與討論：

(1) K-means 在各個 n_cluster 以及 n_components 的組合上表現皆不佳，推測是街景分類並沒有相對明顯的特徵能夠進行分群，K-means 沒有能力處理較為複雜的任務。

(2) 實作時，決定每一群資料 label 的方式是選擇該群中最多的 label，在 $n_cluster$ 接近 class 的數量時，可能會發生某個 class 不會對應到某一群的情況，因而輸入該 label 時，不可能會得到正確答案。(即 confusion matrix 中某一個 column 的情形)

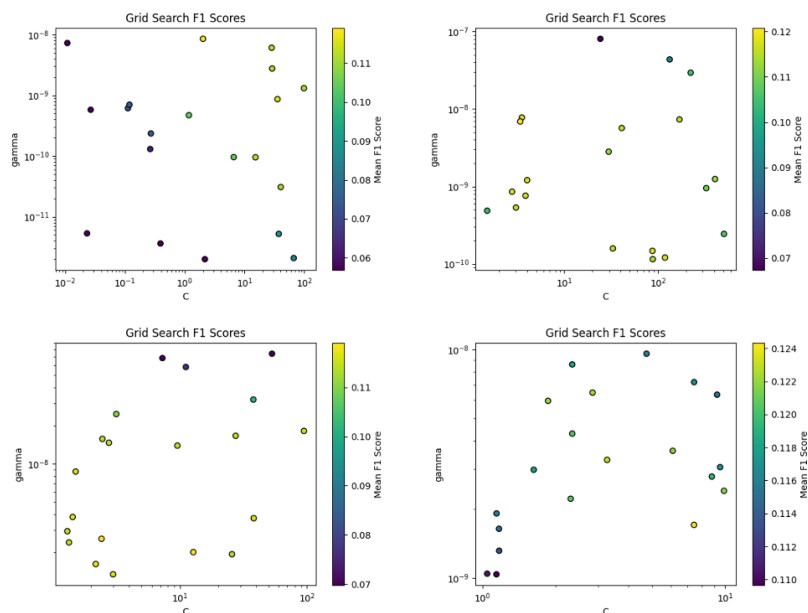
(3) 由於 K-means 實際上效果並不佳，PCA 對準確度的影響留待後續實驗進行討論。

SVM

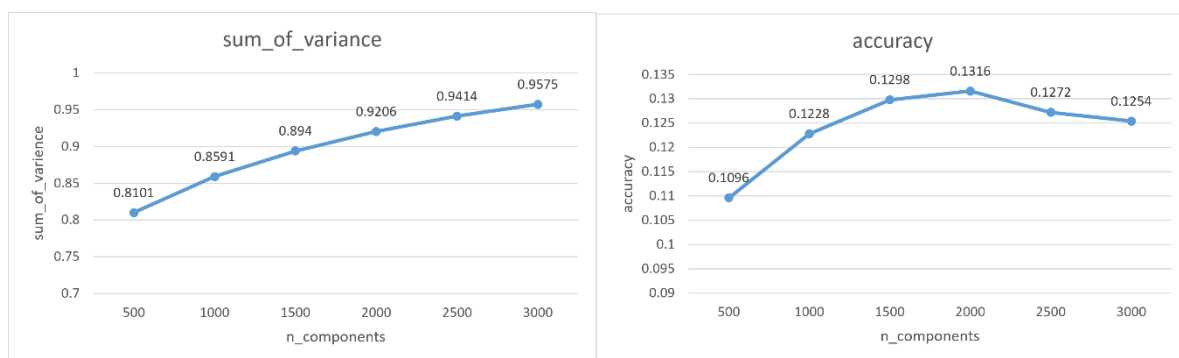
實驗會先使用 PCA 將圖片降維，再使用 SVM 進行訓練及預測。主要調整參數有 SVM 的 C 值、gamma 值以及 PCA 的 feature 數量。

1. 實驗結果：

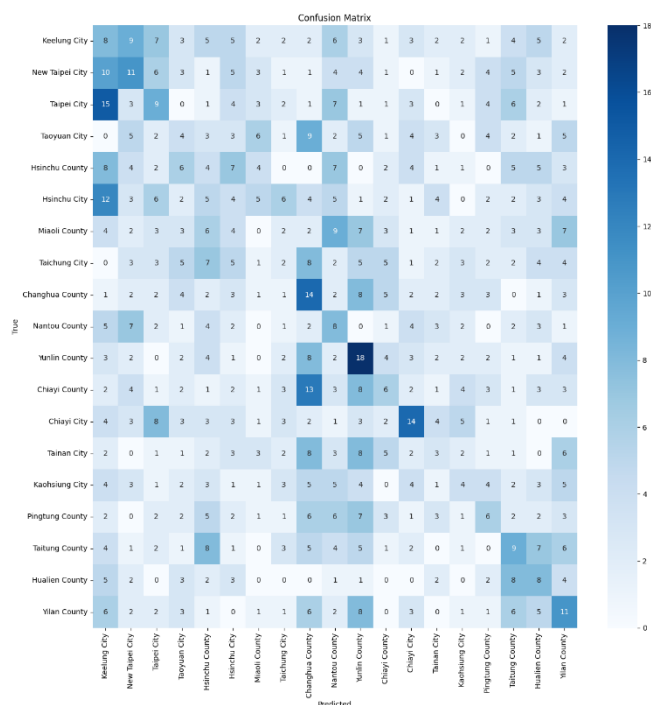
(1) 固定 PCA $n_components=1000$ ，利用 RandomizedSearchCV 隨機選取 20 個 C 值與 gamma 值的組合，並且進行 cross validation。



(2) 取上個實驗中 accuracy 最高的 C 值與 gamma 值，觀察 PCA n_components 的影響。(C = 7.44455、gamma = 1.7083e-09)



(3) confusion matrix :



2. 分析與討論：

(1) 相對於 K-means，SVM 在準確度的表現相對較好。推測因為 SVM 是相對較複雜的模型，能夠擷取的特徵相對較多，因此表現較好。

(2) confusion matrix 中 label 的排序依據地理位置由逆時針排序，並且發現顏色相對深的方格大多聚集在對角線附近，代表模型有學習到各區域的特徵，但分類的精度不足，導致 accuracy 較低。

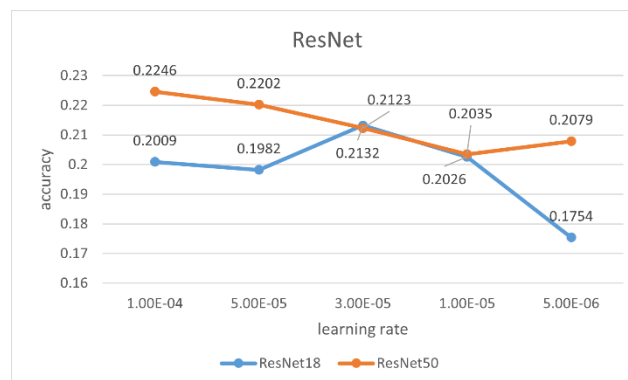
(3) 在 PCA 的實驗中，accuracy 隨著 $n_components$ 的數量先升後降，符合最初的預期。在 $n_components$ 較低時，因為 PCA 獲得的特徵較少，模型較難學到資料的差異。在 $n_components$ 過高時，可能造成 Curse of dimensionality，即維度過高導致資料間的差異性過大，讓模型較難學到相關性。

ResNet

使用 ResNet 進行實驗。實驗會直接讀入圖片，並且進行訓練及測試。經測試後發現沒有 pretrained 的模型效果普遍不佳，因此本實驗皆使用 pytorch 的 pretrained 模型。主要調整參數有 ResNet 的層數以及 learning rate，loss function 使用 CrossEntropy，optimizer 使用 Adam。最終比較會以 validation loss 最低的模型作為該次訓練的最終模型。

1. 實驗結果：

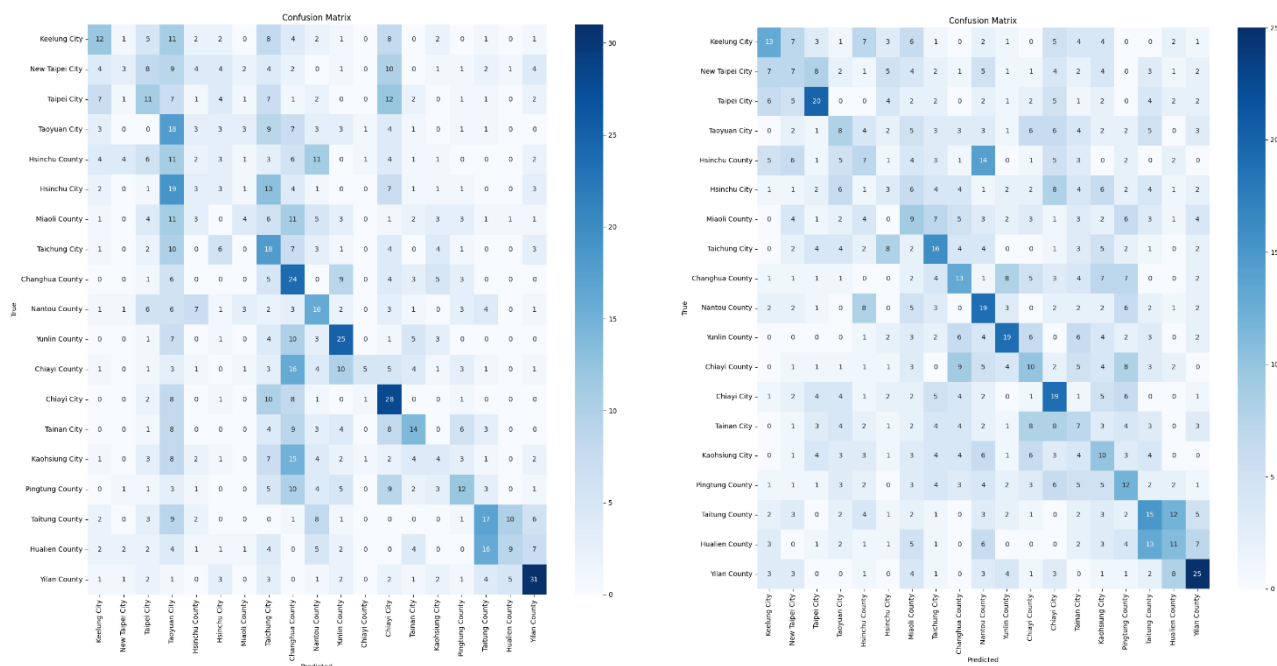
(1) accuracy：



(2) loss graph (此處展示 ResNet50@lr=1e-4(左)、ResNet50@lr=1e-5(右)，其餘 loss graph 請見附件)：



(3) confusion matrix (此處展示 ResNet50@1e-4(左)、ResNet18@3e-5(右)，其餘 confusion matrix 請見附件)：



2. 分析與討論：

(1) 比較 ResNet18 以及 ResNet50 可以發現 ResNet50 大多優於 ResNet18，代表增加模型層數對模型的準確度有一定的提升。

(2) 觀察 loss graph 可以發現，在 learning rate 較高時，模型不需要太多 epoch 就可以完成訓練。相對的，當 learning rate 較低時，就需要跑較多輪訓練。以 ResNet50@lr=1e-4 的 loss graph 為例，模型在 epoch = 1 就已經達到最低的 Validation loss，隨後 Validation loss 趨勢上升，但 training loss 下降，推測有 overfitting 的問題，可以透過 dropout、增加 Dataset 等方式解決。

(3) 觀察 confusion matrix 可以發現，大多數顏色較深的方格聚集在對角線附近，與 SVM 的結果類似。

Dataset

經過以上實驗後，以下對 dataset 進行一些討論：

1. 台灣各縣市的街景與地理特徵存在差異，並且有連續性的變化，模型也有能力分辨這些特徵。可以由依地理位置排序的 confusion matrix 看出。
2. 模型在東部地區表現較好，推測是東部地區的地理特徵相對明顯。在西部地區的部分僅能大致區分北中南，並且在各區域的精確度都較為不足。
3. 在模型的訓練中有出現 overfitting 的問題，可能原因之一是 dataset 資料量不足，可以考慮增加訓練資料。

Conclusion

1. K-means 的模型複雜度不足，不足以達到街景分類的目標。
2. SVM 可以透過調整參數的方式，提升模型的表現。
3. PCA 可以降低圖片維度，加速模型的訓練以及提升模型的表現，但需要選擇適中的 $n_components$ 。
4. ResNet 在增加層數後，能夠提升模型的表現。在訓練部分，應該根據模型的複雜度以及 learning rate 調整對應的 epoch 數量。
5. Dataset 中，台灣的街景存在地區差異，並且有連續性變化，但在縣市區分上應該要有更多的資料才能做到精確度更高的分類。

Future Works

1. 這次實驗主要是以分類問題的形式進行訓練，但實際上也可以使用經緯度作為 label，進行 regression 的訓練。
2. 考量到時間以及成本，本次作業 Dataset 的資料量並沒有太多，未來若有機會可以增加資料集的大小，並且增加更多特徵較不明顯地區的資料。

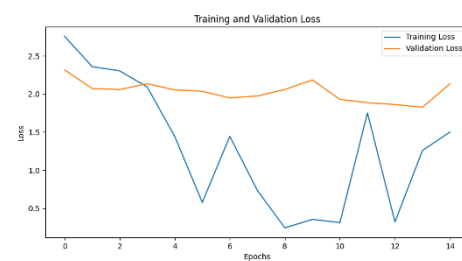
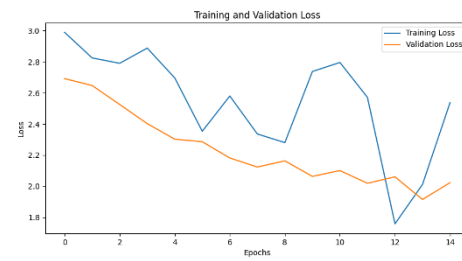
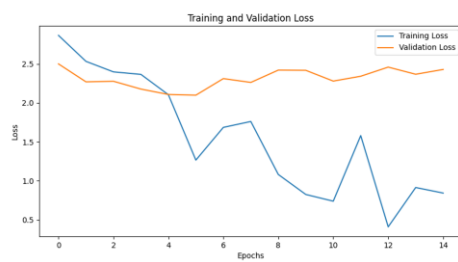
3. 參考現有圖片預測地理位置的模型，在訓練分類問題時，可以將原本 label 的 one-hot vector，依據地理位置給予各分類不同的權重。例如資料在縣市邊界上，label vector 中兩縣市的權重就會接近。以此可以給予模型對於特徵之於縣市的重要性差異。

References

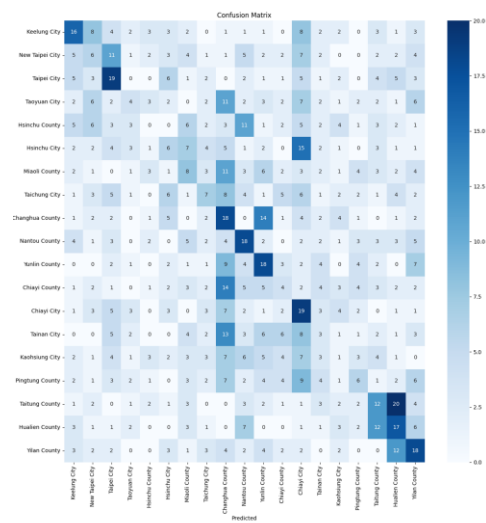
1. scikit-learn : <https://scikit-learn.org/stable/>
2. pytorch : <https://pytorch.org/>
3. Google map API : <https://developers.google.com/maps?hl=zh-tw>
4. PIGEON: Predicting Image Geolocations :
<https://arxiv.org/abs/2307.05845>
5. geoguessr : <https://www.geoguessr.com/>

Appendix

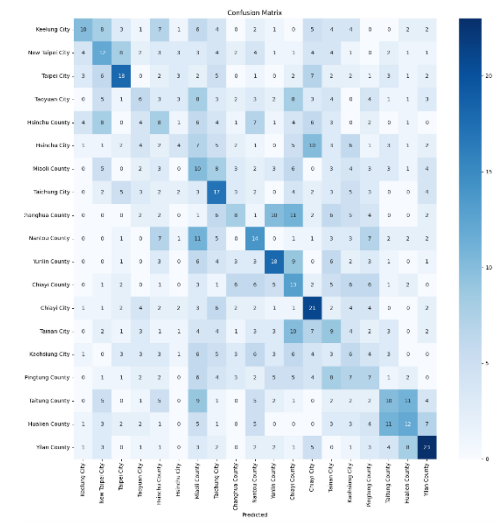
ResNet loss graph



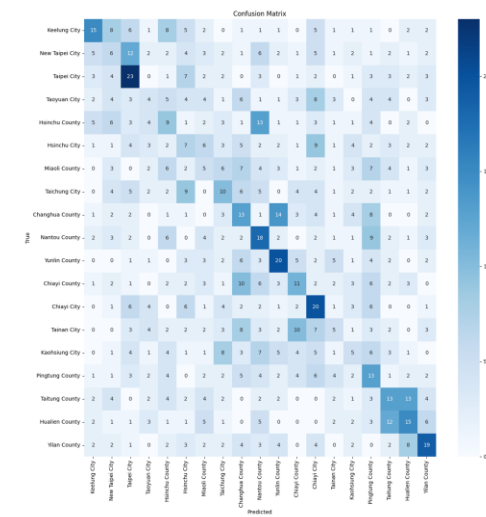
ResNet confusion matrix



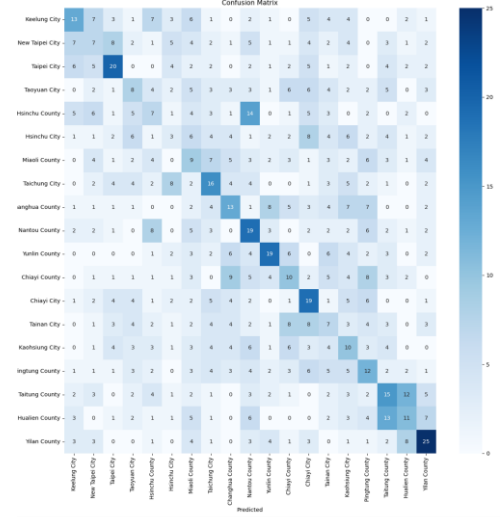
ResNet18@lr=5e-6



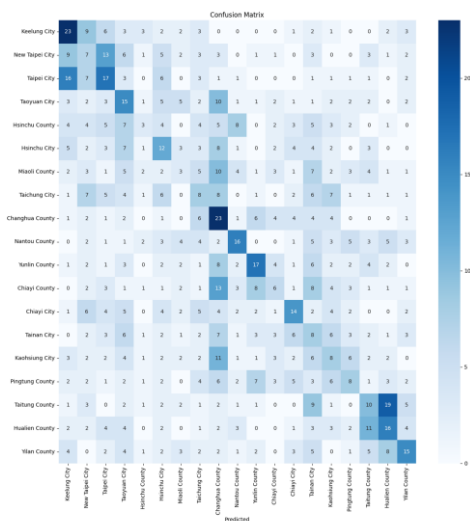
ResNet18@lr=5e-5



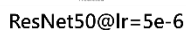
ResNet18@lr=1e-5



ResNet18@lr=3e-5



ResNet18@lr=1e-4



Code

避免版面過於雜亂，本次作業相關的程式碼會放在 github 上。

Link : <https://github.com/yyh-603/google-street-view>