

## 方法描述

這次的作業是要各使用七種擷取特徵的方法(raw image, color histogram, local binary pattern, co-occurrence matrix, gabor filters, histogram of oriented gradient, bag of features)將資料集中的圖片特徵取出來，再使用 nearest neighbor classification 的方法將圖片做分類，我使用的是 SAD：取此張 testing 圖片特徵值，並一一與 training 所有圖片特徵值做相減，並取絕對值，找出 testing 與 training 差距最小的 training 圖片，再將此 training 圖片所屬的植物類別也設定成此張 testing 圖片的植物類別。

在最一開始，先用 `resize_224.m` 這個程式將 training 和 testing 的所有圖片 resize 成 224\*224 的大小。

資料集的處理上有兩種做法：

將一半的 training 資料集 4750/2 張規劃成 validation 資料集，剩下的一半就是要被訓練的 training 資料集。在此 SAD 找出與一張 testing 圖片最相似的 training 圖片後，會檢查兩張所屬的植物類別是否一樣(檢查是否判斷正確)，若一樣則代表 SAD 判斷正確，`correctSAD`(我的程式碼中的變數)會累加一分，最後就能算出 SAD 的準確率。

另一種做法是，training 資料集就是 4750 張，testing 資料集就是 794 張，與 kaggle 預設的一樣。在做完 SAD 的絕對值相減並找出差距最小的圖片類別後，就將此張 testing 圖片的植物類別設定成此 training 圖片所屬的植物類別，並另存在 file 與 species 的陣列(我的程式碼中的陣列)，最終把兩個陣列輸出成 csv 檔案的形式，產生出一個 csv 檔案，再傳上 kaggle 看得到的辨識成績如何。

## 執行方式

在 `plant-seedlings-classification` 這個資料夾中，有 `train` 資料夾、`test` 資料夾以及七種擷取特徵方法的資料夾，還有三個資料夾名稱句尾有 `_csv` 的代表是能產生 csv 檔案的程式碼。

以 `bagoffeature` 這個資料夾舉例：

資料夾裡有 `loading.m` 和 `bagoffeature.m` 兩個檔案，只要執行 `bagoffeature.m` 這個程式就可以得到所要的 SAD 辨識答案。

執行 `bagoffeature.m` 後，在程式第三行會呼叫 `loading function`，以取得 `trainMatrix`(training 的各圖片特徵)，`testMatrix`(testing 的各圖片特徵)，

trainBelong(training 的各圖片對應到的植物類別), testBelong(testing 的各圖片對應到的植物類別)。

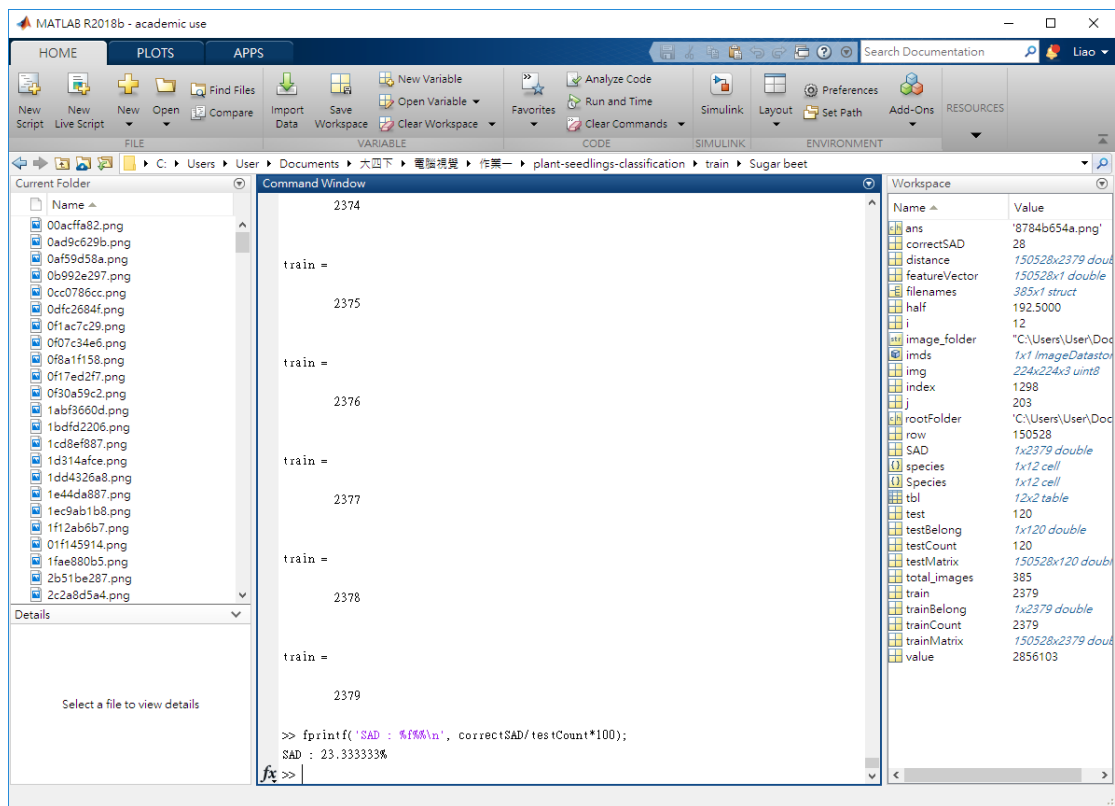
在 loading.m 中, 會一一進入到 training 資料集中各個植物類別的資料夾, 以提取各資料夾中圖片的特徵。

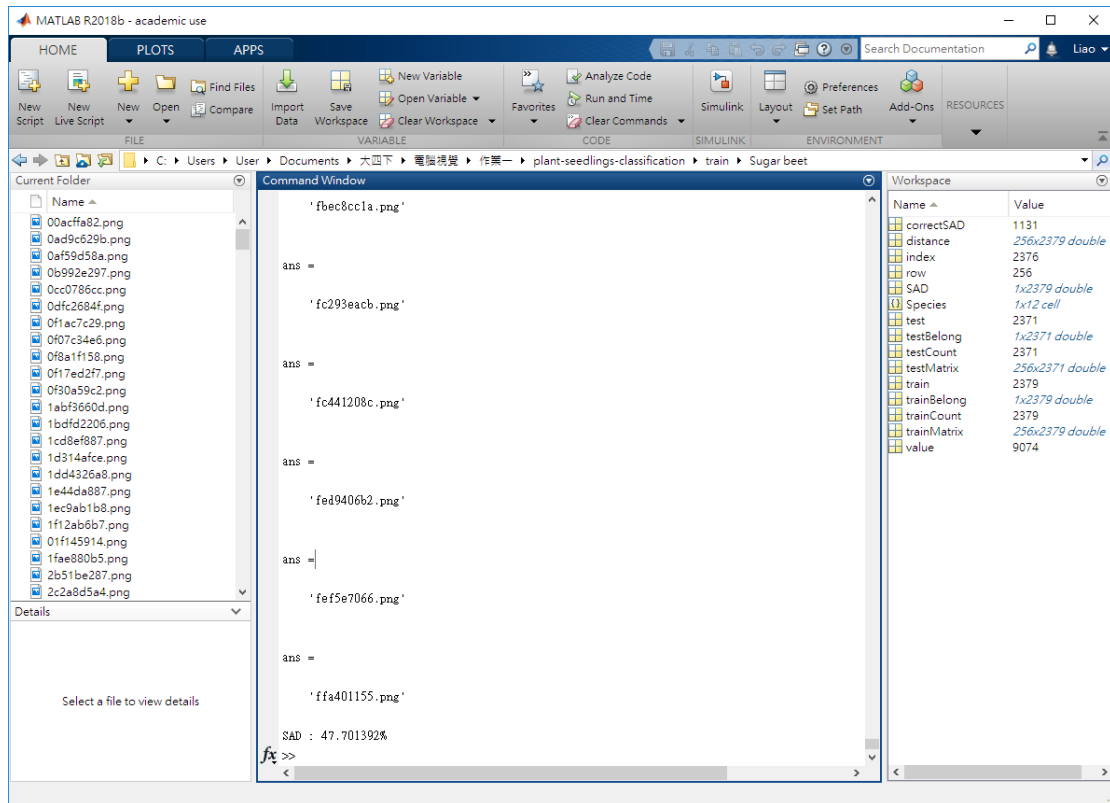
回到 bagoffeature.m 後, 將每個 testing 的圖片特徵一一與 training 的所有圖片特徵做相減, 再找出相減後取絕對值最小者。再檢查兩張所屬的植物類別是否一樣(檢查是否判斷正確), 若一樣則代表 SAD 判斷正確, correctSAD(我的程式碼中的變數)會累加一分, 最後將 correctSAD 除以 testing 的總數再乘以 100, 算出 SAD 的準確率。

若是要執行可以得到 csv 檔案的資料夾, 最後產生的資料夾會在 plant-seedlings-classification 這個資料夾中。

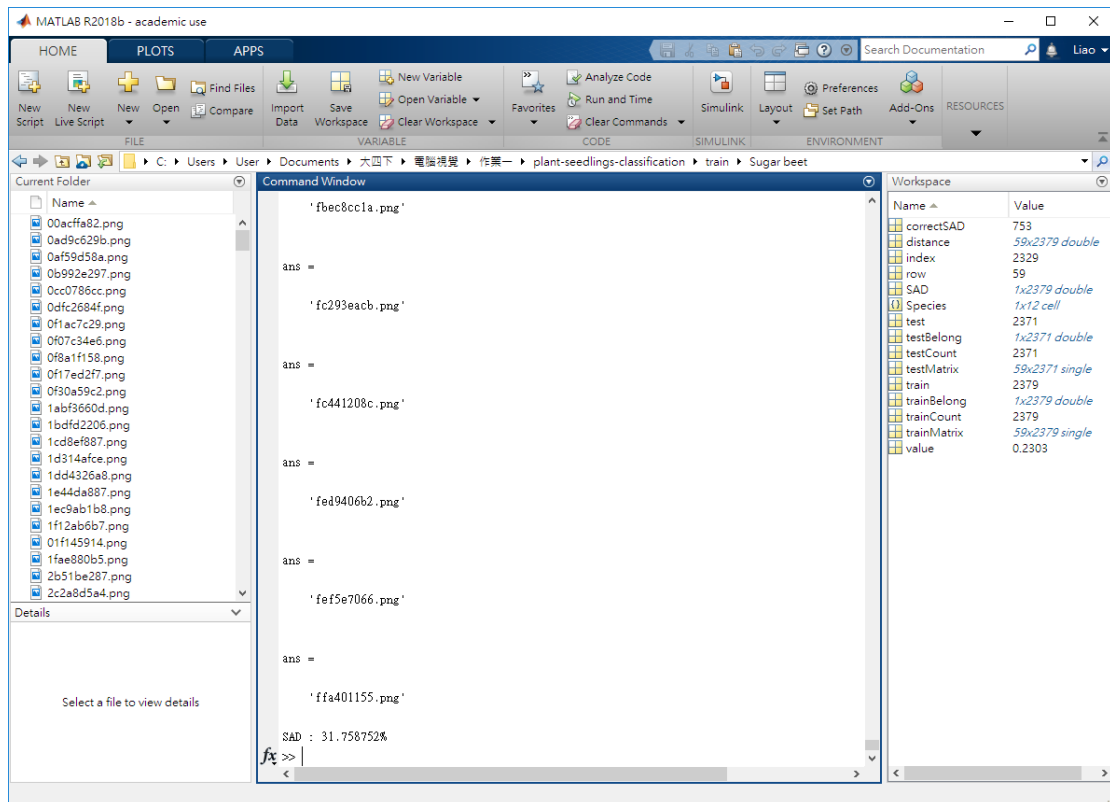
## 實驗結果

training 劃分一半為 training 資料集，另一半劃分為 validation 資料集：

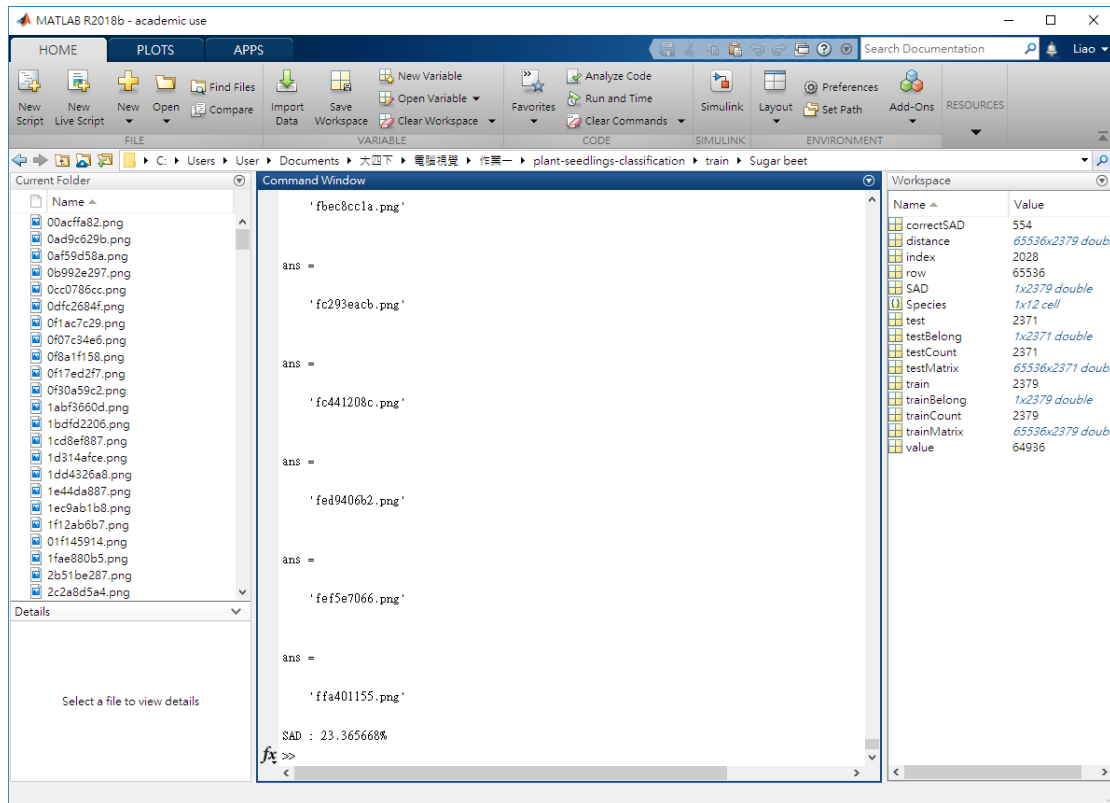




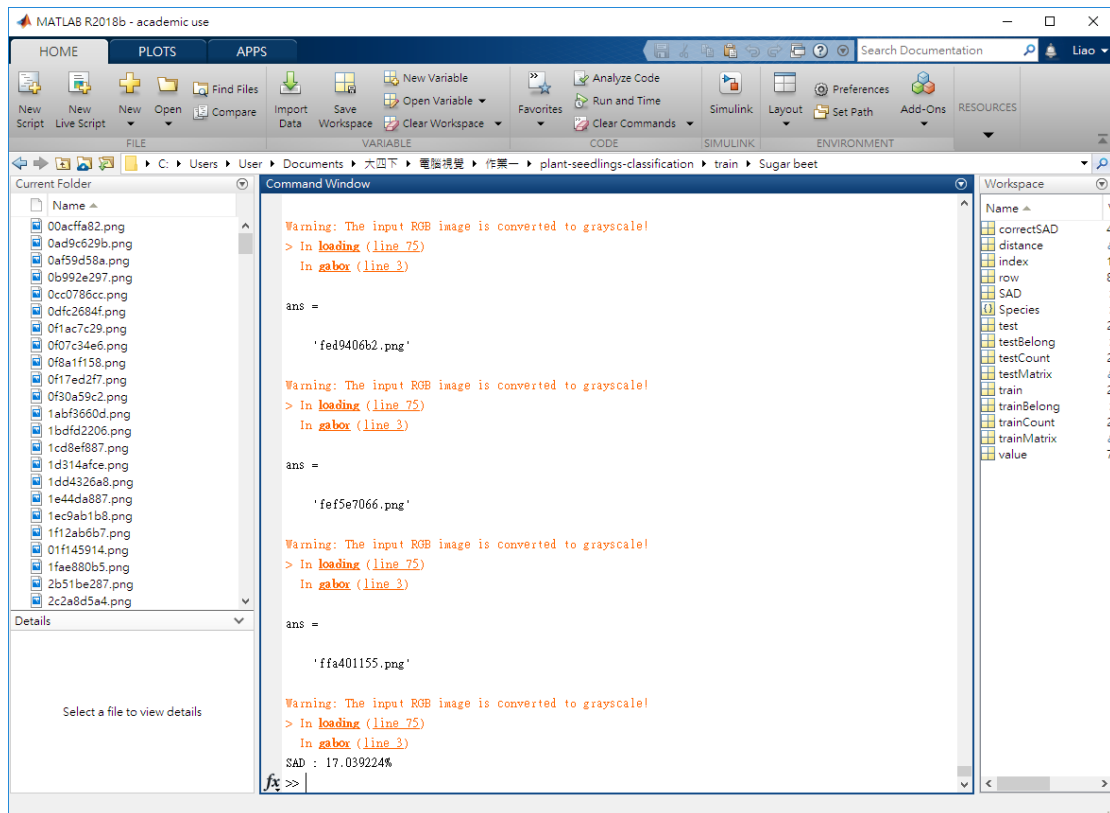
color 準確率 47.701392%



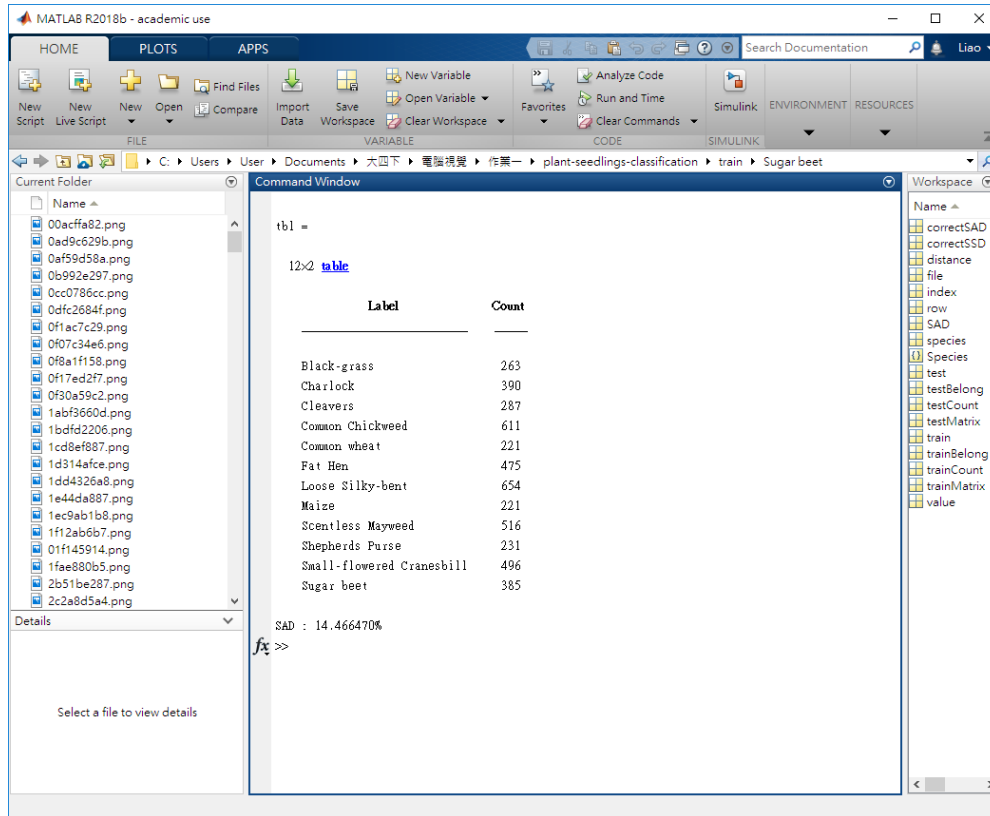
lbp 準確率 31.758752%



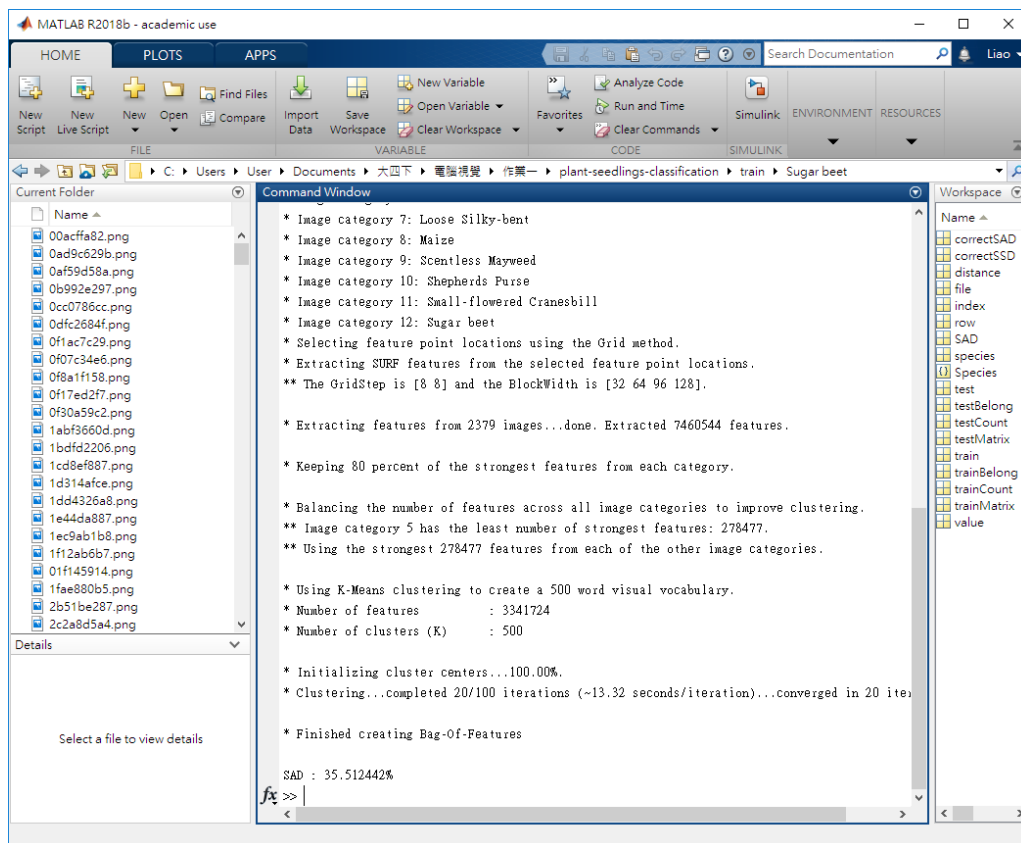
co-occu 準確率 23.365668%



gabor\_filter 準確率 17.039224%




hog 準確率 14.466470%



bagoffeature 準確率 35.512442%

按照 kaggle 規定的 training 資料集、testing 資料集：



Plant Seedlings Classification  
Determine the species of a seedling from an image

Kaggle · 836 teams · a year ago

Overview Data Kernels Discussion **Leaderboard** Rules Team My Submissions **Late Submission**

Your most recent submission

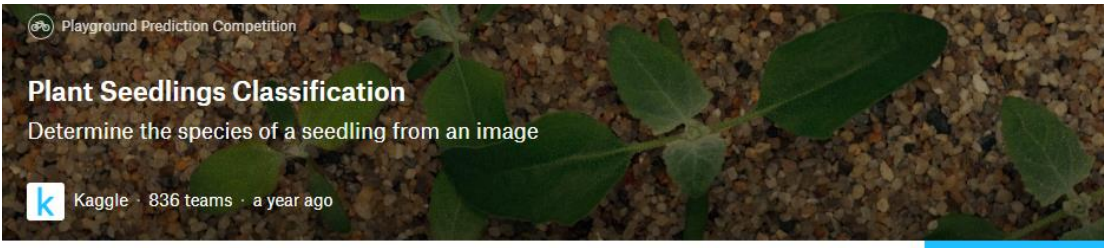
Name	Submitted	Wait time	Execution time	Score
submission_bag.csv	just now	1 seconds	0 seconds	0.36964

Complete

[Jump to your position on the leaderboard](#)

bagoffeature

第 784 名



Plant Seedlings Classification  
Determine the species of a seedling from an image

Kaggle · 836 teams · a year ago

Overview Data Kernels Discussion **Leaderboard** Rules Team My Submissions **Late Submission**

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission_co.csv	just now	0 seconds	0 seconds	0.20780

Complete

[Jump to your position on the leaderboard](#)

co\_occu

第 794 名



Playground Prediction Competition

## Plant Seedlings Classification

Determine the species of a seedling from an image

Kaggle · 836 teams · a year ago

Overview Data Kernels Discussion **Leaderboard** Rules Team My Submissions **Late Submission**

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission_color.csv	just now	0 seconds	0 seconds	0.53148

Complete

[Jump to your position on the leaderboard](#) ▼

color

第 767 名(最高)



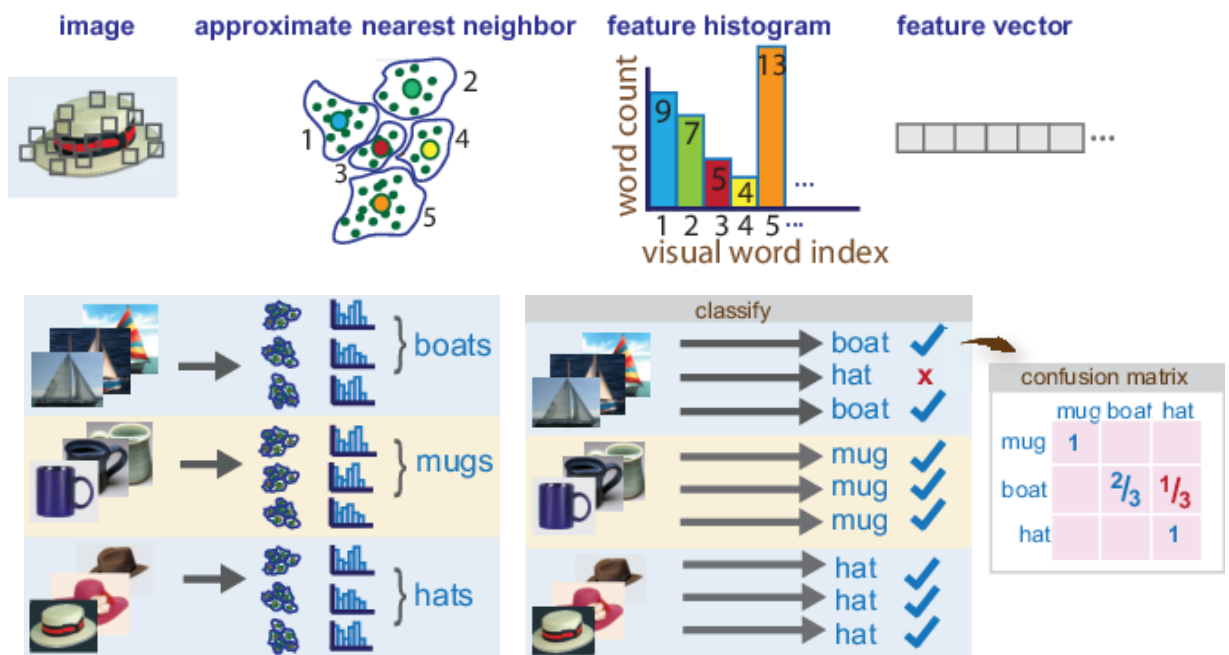
## 結果討論

training 資料集 2375 張，validation 資料集 2375 張

擷取特徵方法	raw image	color histogram	local binary pattern	co-occurrence matrix	gabor filters	histogram of oriented gradient	bag of features
準確率	23.333333%	47.701392%	31.758752%	23.365668%	17.039224%	14.466470%	35.512442%

準確率最高的是 color histogram，次高的是 bag of features。

在還沒開始寫作業之前，我以為準確率最高的會是 bag of features，因為這個方法內部做了最多步驟，如下圖



但是 color histogram 的準確率結果更好。我想是因為這次的所有圖片，拍攝的亮度幾乎一樣，而且拍攝角度皆是從植物正上方往下拍，使同類別的植物圖片一致性很高。color histogram 是直接取出紅、綠、藍在各圖片中所佔的位置、份量，綜合上述這次資料集的特性，才會使 color histogram 的表現較好。

## 問題與遇到的困難

第一個遇到的困難是讀取資料夾的路徑與名稱。因為要先進入 training 資料集，再一一進入各個植物類別的資料夾，才可以進行提取特徵的步驟。一開始對第一個植物類別 Black-grass 資料夾中的所有圖片提取特徵後，程式會停住，錯誤訊息說找不到對應的路徑名稱，後來發現是因為還停留在 Black-grass 資料夾內，在 Black-grass 資料夾內當然找不到下一個植物類別 Charlock 的資料夾，所以跑完每層資料夾後，把目前位置用 `cd..\` 就可以回到 train 資料夾的位置，並順利找到下一個植物類別的資料夾。

第二個遇到的困難是，執行 raw image 方法時，嘗試無數次，我的筆電都會從連滑鼠游標都不能移動，到最後直接藍色螢幕自動重新開機。因為我是從投影片作業要求的 bag of features、hog、……一個一個往前實做出來，所以一開始我覺得既然其它六個方法我都可以跑得出結果，唯獨最原始方法的 raw image 會一直遇到藍色螢幕，真的非常奇怪。

推測是因為 raw image 是直接取出每張圖片  $224 \times 224 \times 3$  的值，所以每張圖會是  $150528 \times 1$  的陣列被存到 trainMatrix 以及 testMatrix，又因為全部的圖片數量也不少，我的筆電記憶體就顯得不足，才會發生藍色螢幕。

後來詢問老師後，老師說可以縮小 testing 資料集的數量來做測試。於是我慢慢地調整 testing 資料集的圖片數量，最後設計成取每個植物類別各 10 張當作 testing 資料集，才可以在筆電不會當機，且不會過久的執行時間內，可以跑出結果。