**Overview**

這次的資料來源是一個芒果分類競賽所提供的，我們的任務是將彩色的芒果圖片分類到三個等級之中（A、B、C），A 的芒果是品質最好的、B 次之、C 是品質最差的。以下將會先對資料做簡單的分析，並接著進行這次模型的訓練分析。

**Data Analysis & Observations**

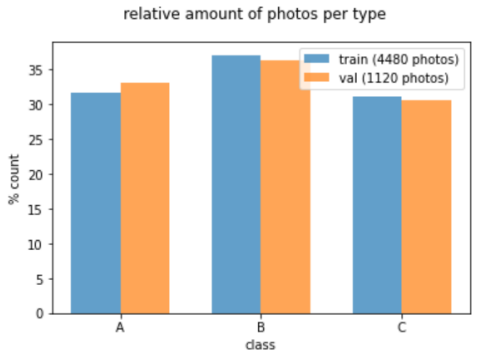
這次的分類跟以往的 lab 比起來比較複雜，我們不是做一般的 binary classification，而是有三個 class 的 multi-class classification，在假設資料分布平均的狀況下，瞎猜命中的機率就從 50%降到 33%，外加接下來會探討到的影像特質等，都給訓練過程帶來了更大的挑戰。所以有鑑於這次分類的難度增加，我覺得對資料先進行初步的評估和分析，在接下來的建模中，或許可以幫助我朝比較好的方向邁進，也說不定可以少走些冤枉路，抑或對訓練結果做出比較好的解釋。

首先，看一下三類中芒果大概長的樣子：

| | | |
|---|---|---|
| 等級 A |  |  |
| 等級 B |  |  |
| 等級 C |  |  |

可以看到資料集和之前的芒果顏色分類（分類綠和紅兩類）相似（應該是源出自同一個資料集），影像的大小都不依，比例不同，背景也有很多干擾（握芒果的手、裝其他芒果的籃子、雜物、不同顏色背景等）。仔細看一下上面的幾張範例，可以大概看出，等級的評比不單只和顏色有關係：紅潤完善的芒果當然就是 A，沒有甚麼斑紋但是顏色有點帶綠的芒果是 B、顏色不一定很綠，但是明顯帶有不新鮮斑紋的芒果是 C，所以分類的一句大約分成兩大方向：**顏色**和**花紋**。

我切的 training set 和 validation set 資料分布如下，可以看到，大致上上三種芒果的分類都蠻平均的，就 B 等級的占比多了一些，但整體而言，training 和 validation 的資料都沒有很嚴重的 skew，訓練過程中比較不用去擔心這一點。




**Reference & Relevant Literatures**

在進行訓練前我有收集了一些資料，閱覽了幾篇論文，有兩篇我覺得對這次的 lab 最有幫助的如下，將會做簡單的內容簡介：

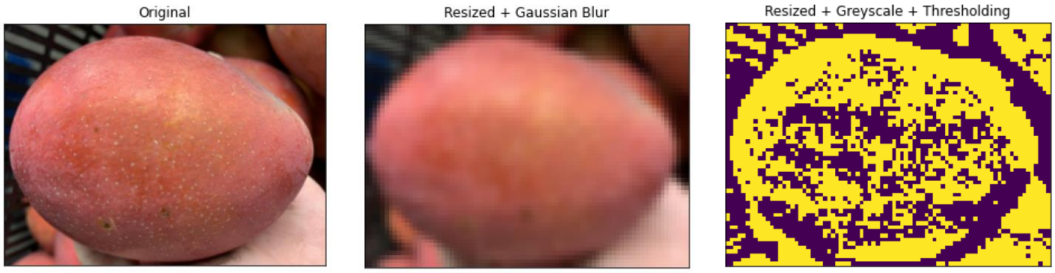
1. **Embedded Machine Learning for Mango Classification Using Image Processing and Support Vector Machine (Minh Thanh Vo et al.,2019)**
2. **Machine Vision Based Fruit Classification and Grading – A Review (Naik et al.,2017)**

（一）第一篇論文是一個越南的研究團隊，為農民開發個用樹莓派板子，對他們當地芒果品種做優劣分類，包括拍照、辨識、分類，主要是依據芒果的大小去估測他的重量然後依這個做分類，分類是用 SVM 作為 classifier，然後判斷大小是用 blurring 外加 thresholding 的前處理將芒果和背景分離，取得芒果的大致形狀。

（二）第二篇論文綜合了並探討了各式各樣水果的分類，也依水果的外觀、大小去作品質分類，這篇論文的價值在於它利用了了很多前處理的手法，包括先前提到的 blurring、thresholding，還有 HOG (Histogram Oriented Gradient) 以及 LBP (Local Binary Pattern)，外加很多不一樣的模型像是 SVM、KNN、Decision Tree、PCA 等，簡單介紹一下論文提到的 HOG 和 LBP：
HOG 是一個 feature descriptor，它會依造影像中的梯度變化去偵測物體的邊邊角角，是一個影像辨識中蠻常用的工具。
LBP 也是一個 descriptor，但它是根據影像的每個像素去和鄰近的像素做比對，常用在於花紋紋路的判別和分類。

**Data Preprocessing & Feature Engineering**

一開始我先按照找到的資料做一些標準的前處理，包刮 Gaussian Blur、Adaptive Thresholding，以及先前提到的 LBP、HOG，以下範例大概呈現處理完的樣貌：




Resize 是必須做的，在訓練的過程當中我原先只拿一部份的資料做訓練（因為資料量龐大，讀取都要花很多時間，所以一開始只拿幾個評估模型效能），而也簡單統計影像比例，大部分都是 4:3 的比例，所以就按照這樣 resize 成 400 x 300，但在後期的訓練階段發現這樣資料讀近來在做 PCA 的時候 memory 會不夠，所以還是必須保守一點，用 80 x 60 的比例，而事後比較，其實影像大小對我所用的模型影響並沒有很大，只要夠讓後面的矩陣運算塞進記憶體就好了。

**Improve Performance**

這次的 Lab 因為沒有要求手刻，所以在建模上我就有更多時間和更多機會去嘗試調比較複雜的模型（像是 SVM 的 kernel 真的有點難刻上次就放棄了）。突然有了這麼大的自由度讓我十分興奮，也真的能理解為甚麼有強大的 library 對寫程式和做研究帶來多大的幫助。我會分別討論一下我嘗試的模型，利用不同的模型來試圖增加自己的 performance，以及 PCA 的使用和參數的設定。

- **Normalization、Dimensionality Reduction & Fitting Data**
  在進行 PCA 前一定要先將資料 normalize，還有 PCA 及 normalization 用在 train 以及 test/val 上是有些微的不同：PCA 以及 normalize 都只能 fit 在 training data 上，再將其 transform，對於 test 和 validation 就只能 transform，不能 fit，因為對我們來說它們是未知的資料，這些未知的資料分布特性不應該出現在我們的訓練當中，不然就有點是在「偷看」，可能會影響到訓練的結果和可性度，這一點非常重要，也是我一開始沒注意的，後來才想到做了正確的更動後準確度也很神奇的上升了 1 ~ 2 %，所以這可能也是當初要我們手刻的意義，因為知道模型和資料處理背後的原理，我們在 handle 自己的 data 才會比較謹慎，做出的模型才正確。
  另外，PCA 的 ”n_components” 參數我將其設為 0.95，這樣子可以確保每個維度對資料分布的解釋性，就不用花很多不必要的時間和心血去找到底要留下幾個維度。

- **Models**

這次的訓練過程我用了不少模型去訓練我的資料，結果如下表：

| Model/Accuracy | Validation Set | Kaggle Testing Set |
|---|---|---|
| SVM | 54.642857 % | 54.000 % |
| LR | 56.160714 % | 56.250 % |
| KNN | 58.392857 % | 57.999 % |
| **XGBoost** | **64.910714 %** | **64.125 %** |
| Ensemble (Major Voting) | 60.267857 % | 60.000 % |

可以看到 SVM、LR、KNN 的準確度都落於 54 ~ 59% 之間，就只有 XGBoost 有顯著的準確度提升到 64 %，值得一提的是，在做 XGBoost 的時候我並沒有先對資料進行 PCA，和其他模型不同的地方是，XGBoost 不做 PCA 準確率反而會上升，差距會有 3 ~ 4 %。

最後我還建了一個簡單的 Ensemble，利用 Major Voting 的方式，針對所有模型預測結果去做 polling 決定最終預測，可以看到，雖然準確率是可以上六十，但是依然表現還是沒有 XGBoost 好。

**Difficulties Encountered and Solved**

其實後來用了那麼多前處理的技巧發現對 training 都沒有很大的幫助，令人沮喪的是，甚至跟未處理只有 resize 的影像差不多，頂多在準確度上好個 0.5 %，回想起一開始做的資料分析：

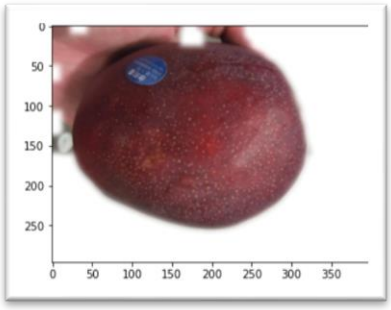
在先前資料分析的時候有討論到芒果判斷的兩個大致依據，所以在對圖片做影像處理的時候可能較要用比較可以擷取這兩個特徵的手法，但其中可能會遇到的干擾有：

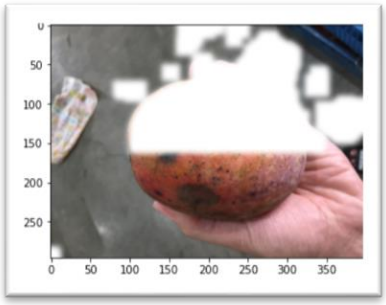
- 背景顏色影響芒果判斷
- 背景形狀影響芒果判斷

用 HOG 判斷型態但也會將背景的東西考慮進去、用 LBP 擷取材質特徵但有時握芒果的手也會影響，我在做前處理一直無法像先前提到的兩篇論文那麼成功我覺得最主要的原因在於背景雜訊太多，有些芒果又是直接握在手上，所以很難過濾掉，而兩篇論文的芒果都是在單一顏色的素背景拍攝，可以輕易地排除雜訊。我有**試著手刻一些過濾器但效果都沒有很好**，可以由下圖兩個比較極端的例子看出效果良莠不齊，除非芒果有很鮮豔的顏色和背景對比，否則很難切的乾淨，甚至會有將芒果切掉的可能。

**原圖**


**過濾後**






所以後來決定影像前處理的時候，能讓芒果在圖中的佔比愈大應該能愈有效的減少 noise。針對這點我就用 crop 的方式，直接用 indexing 來擷取影像範圍，果不其然對訓練有了大大的幫助。

總結這次 Lab，提升準確度最大的兩個因素主要應該落於影像的前處理（濾掉芒果背景雜訊）還有找到適當的模型（從前面的實驗結果就可以看到好的模型塞選是會帶來很大的影響）。第二點我這次的實驗總算是有做到，但至於前處理的地方雖然整體總共花的時間和心血都比建模型上來的多好幾倍，但我覺得還是有進步的空間，我已經竭盡自己所能也找了很多網路上的資料，要在 Python 中做出好的過濾器，尤其對背影那麼複雜的資料我覺得不大容易，這或許也是這個比賽最大的挑戰之一吧。