# Rotten Fruits Classification

Group 29
組員:0710008 曾訪晴 0710025 柯婷文

## Github repo link

https://github.com/clairetsengeecs/NYCU-2021Spring-Introduction_to_AI

## Introduction

隨著人工智慧的蓬勃發展，影像辨識技術被廣泛地應用，從生活中常見的人臉辨識、車牌辨識，到協助醫生分析醫學成像及在工廠中辨識不良品，影像辨識已成為一個非常重要的領域。我們希望透過這份研究，結合這一個學期在人工智慧概論及影像處理概論這兩門課所學的知識，初步探索影像辨識幾個常用的演算法。我們從 kaggle 上挑選了 Fruits fresh and rotten for classification 這個資料集，訓練 CNN 模型及 SVM 模型辨識蘋果、橘子、香蕉，以及判斷是新鮮的還是腐爛的水果。但由於 kaggle 資料集提供的圖片大多是單一背景，大多圖片裡也只有一個物體，而現實世界的圖片卻常有複雜的背景，物體也可能擺放成各種方向，故我們在網路上搜尋相關圖片，用分類器判別，和 kaggle 資料集判別的結果比較。

## Related work

在 Lu 和 Weng 的研究 [2]中整理了過去用於影像辨識的演算法，並且提到可以用 supervised 或 unsupervised、parametric 或 nonparametric、hard 或 soft (fuzzy) classification、per-pixel 或 subpixel 或 perfield 等方式去分類。其中 Supervised 方法包含 Maximum likelihood、minimum distance、artificial neural network、decision tree classifier 等等演算法，Non-parametric classifiers 包含 Artificial neural network、decision tree classifier、evidential reasoning、support vector machine、expert system 等等。我們的研究從以上挑選了一些演算法來實作。

在 Li 等人的研究中[1]，建立一個 CNN 模型分類肺部的影像。他們認為這些影像是 texture-like，沒有顯著的結構，太多層 convolution layer 不必要且可能有 overfitting 的問題。因此他們只採用一層 convolution layer。2011 年 Zhang 等人的研究[3]針對 SVM 分類器做討論，實驗數據顯示圖片前景的特徵並不會提升分類器的表現，真正主宰物體辨識的還是物體的特徵。然而，使用背景過度單一的圖片作為訓練資料集時，在複雜的測試資料集上會表現不佳。

而在 Engstrom 等人的研究裡[4]，討論了圖片背景在機器學習模型裡扮演的角色。他們使用 OpenCV 來將 ImageNet 裡的圖片做前景跟背景的分離，再進一步產生七種新的資料集，分別是前景用黑色方框取代、前景用其他背景取代、前景變成黑色、背景用黑色取代、背景用隨機的不同類別背景取代（Mixed-Rand）、背景用下一個類別的背景取代、背景用相同類別的其他背景取代的圖片。實驗結果發現，使用 ImageNet（正常圖片）來訓練的模型在預測 Mixed-Rand 的圖片時，準確率會下降 6-

14%，可見模型在預測時的確有運用到背景的資訊。若是使用 Mixed-Rand 來訓練，則可以提升預測具有誤導性背景的圖片時的準確率，也降低了模型對於背景資訊的依賴程度。

# Methodology

我們的資料集分為兩類。首先，使用 kaggle 上的 Fruits fresh and rotten for classification dataset，裡面的水果圖片分成六類，已經拆成 train 和 test 兩個部分，分別為新鮮的蘋果（1693 張 train 加 395 張 test 圖片）、新鮮的香蕉（1581 張加 381 張）、新鮮的橘子（1466 張加 388 張）、腐爛的蘋果（2342 張加 601 張）、腐爛的香蕉（2224 張加 530 張）、腐爛的橘子（1595 張加 403 張）。資料集包含從不同角度拍攝的照片，例如從水果的側面或上方拍攝。資料集中包含加入椒鹽雜訊、旋轉 15 度至 75 度、平移、垂直翻轉等處理過的圖片。由於圖片大小不一，我們先將他們依照需求 resize 成 128*128*3 或 64*64*3 的大小。除此之外，圖片來源是 PNG 檔，包含了 RGBA 四個 channel，由於 Alpha 的影響並不大，故我們只保留 RGB 做計算。

除了 kaggle 的資料集，為了測試我們的模型對一般人拍攝或是在網路上的照片是否有一樣的準確度，我們在 Google 上搜尋「fresh apple」等關鍵字，用爬蟲抓取各類別的照片。經過整理，移除不是那三種水果的照片，例如食譜封面、蛋糕、果汁、插畫等，剩下的圖片裁剪成接近正方形，避免在 resize 的時候過度變形。最終挑出新鮮蘋果（11 張）、新鮮香蕉（15 張）、新鮮橘子（9 張）、腐爛蘋果（8 張）、腐爛香蕉（10 張）、腐爛橘子（10 張）做為測試資料。

第一個演算法我們使用影像辨識最常用的 Convolutional Neural Network (CNN)，第二個演算法我們使用 Support vector machine (SVM)。評估模型的指標我們採用了 accuracy、precision、recall 以及 f1-score。

## 方法一：CNN

*架構*

CNN 模型的架構包含 Convolution Layer 和 Fully Connected Layer，我們在每一層 Convolution Layer 後面加入 Batch Normalization、ReLU 和 Max pooling。實驗中我們多採用三層 Convolution Layer 加上三層 Fully Connected Layer。

*Convolution Layer*

Convolution Layer 好處在於透過影像通常由許多相同的特徵（如特定的線條、輪廓等）組成的特性，只要使用相同幾個神經元組成的卷積核，透過滑動窗口對整張圖片做卷積，便可以在進到 fully connected layer 前大幅減少參數量，降低計算負擔。另一個好處則是卷積可以保留位置資訊，也就是圖片間 pixel 相鄰的關係等。若沒有使用 CNN，直接將圖片 pixel 做 flatten 進入 fully connected layer，則某個維度的相鄰 pixel 在 vector 上必定會相隔一定的距離，如此一來，空間資訊就大幅的消失了。

*Batch Normalization*

Batch Normalization 可以讓數據滿足平均值為 0，標準差為 1，讓數據不會過大、增加穩定性、加速收斂等。公式為$y = \frac{x-mean(x)}{\sqrt{var(x)+eps}} * \gamma + \beta$，其中 eps 預設值為 $10^{-5}$，避免分母為 0，$\gamma$預設為 1，$\beta$預設為 0。

*ReLU Activation Function*

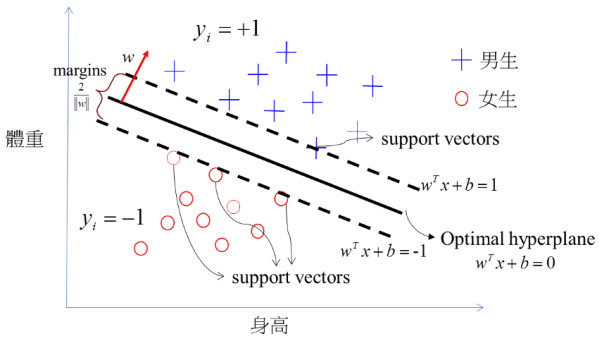
我們選擇 ReLU 作為 activation function，可以有效率降低梯度下降及反向傳播的問題，以及簡化計算過程。ReLU 的公式為$f(x) = max(0, x)$。

*Pooling Layer*

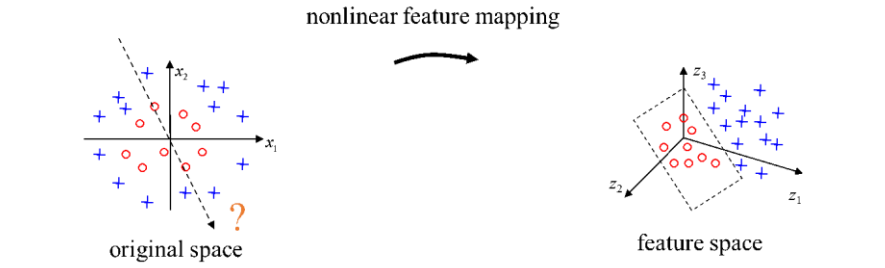
我們採用兩種 max pooling layer，針對不同階段減少計算量。我們的 CNN 模型有 3 層 layer，前兩層使用步幅為 2，池化窗口為 2*2 的 max pooling layer，對每個區塊中的四個數字取最大值，資料量減少為四分之一。最後一層用步幅為 4，池化窗口為 4*4 的 max pooling layer，在 16 個數字中取最大值，資料量減少為十六分之一。

## 方法二：SVM

Support Vector Machine (SVM)是假設有一個 hyperplane ($w^T x + b = 0$)可以完美分割兩組資料，找參數(w 和 b) 讓兩組之間的距離($\frac{2}{||w||}$)最大化。舉例來說，要用身高和體重去分辨是男生或女生，男生是$w^T x + b \geq 1$，女生是$w^T x + b \leq -1$。轉換成數學公式是$max\{\frac{2}{||w||}\} \rightarrow min\frac{1}{2}w^T w, subject\ to\ y_i(w^T x_i + b) \geq 1, \forall i = 1, \ldots, n$。（本段圖片及文字擷取自 機器學習-支撐向量機(support vector machine, SVM)詳細推導）




而有時候在原始空間無法將資料做線性劃分，此時必須利用 kernel，把資料做非線性投影到更高的維度，才能做劃分（如下圖）。而在我們的實驗裡，我們採用 RBF（Gaussian Radial Basis Function Kernel）作為 kernel function，他會將資料轉換成高斯函數的線性組合，好處是可以做到無限多維。（圖片來源：機器學習: Kernel 函數. 在機器學習內，一般說到 kernel 函數都是在 SVM 中去介紹，主要原因是 SVM 必須…）




It's difficult to find the linear separating relationship between this data set.

It's easier to find complete linear separating relationship in the feature space

評估模型

評估一個模型的優劣可以從許多個方面來看，一般常用混淆矩陣(Confusion Matrix)來分析，分成四個部分：(1) 真陽性(True Positive, TP)：預測為 Positive 且預測準確 True，例如預測有下雨且真的下了。(2) 真陰性(True Negative, TN)：預測為 Negative 且預測準確 True，例如預測不會下雨且真的沒下。(3) 偽陽性(Flase Positive, FP)：預測為 Positive 但實際為 Negative，例如預測有下雨實際沒下。(4) 偽陰性(Flase Negative, FN)：預測為 Negative 但實際為 Positive，例如預測不會下雨但實際下雨了。由於我們要將水果分成六類，不是只有陰性陽性，因此我們的混淆矩陣的大小會是 6*6。此外我們還用 sklearn.metrics 中的 classification_report 來分析結果，他會對每一個分類提供 precision、recall、f1-score 的結果，並且計算 Macro-averaging（所有類別的每一個統計指標值的的算數平均）和 Weighted-averaging（給每个樣本決策相同的權重）。以下是我們採用的評估指標的說明：

(1) 準確率（Accuracy）：對樣本正確分類的比率，$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$。

(2) 精確率（Precision）：被分類器判定為陽性的樣本中是真正的陽性樣本的比重，$Precision = \frac{TP}{TP+FP}$。

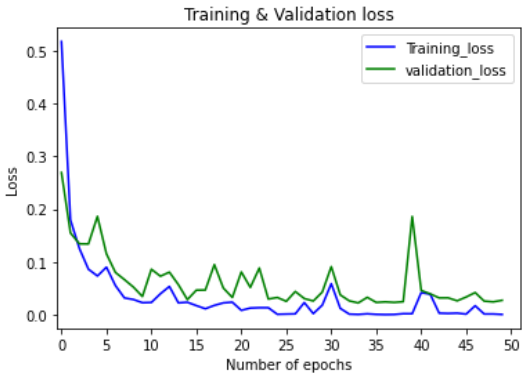
(3) 召回率（Recall）：被分類器正確判定的陽性樣本占所有陽性樣本的比重，$Recall = \frac{TP}{TP+FN}$。

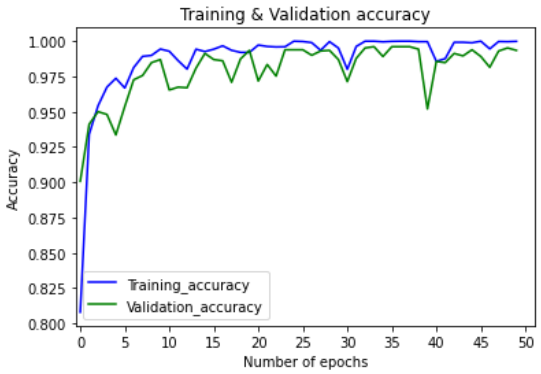
(4) F1-score：precision 和 recall 的調和平均數，$F_1 = \frac{2*TP}{2*TP+FP+FN}$。

# Experiments

實驗一

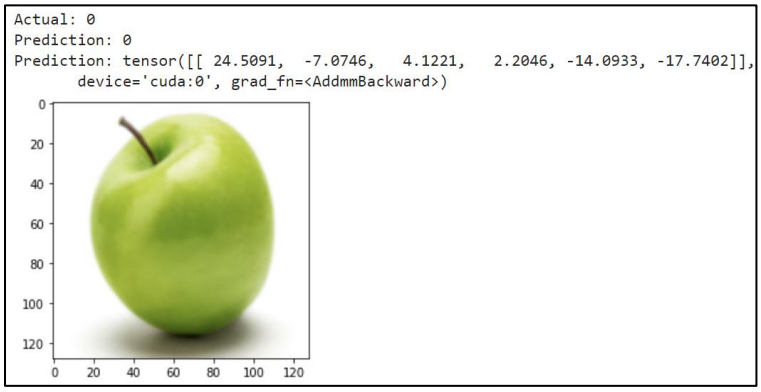
我們將原先屬於 train 資料集的圖片隨機拆成 8:2 作為 training 和 validation。接著用 CNN 建立模型，訓練 50 個 epoch。訓練過程 accuracy 和 loss 隨 epoch 增加的趨勢變化如下圖。



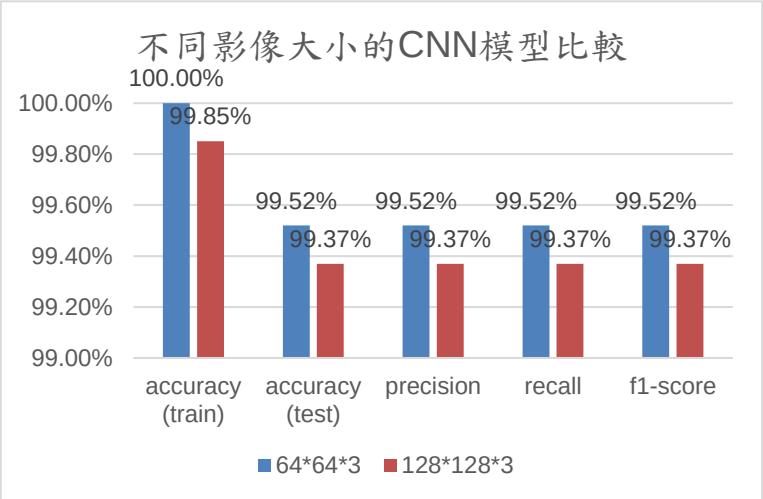

可以看到隨著 epoch 增加，training 和 validation 的準確度先呈現上升趨勢，直到趨近一個極限值。接著拿我們訓練完成的模型去測試，train data 的 accuracy 為 99.85%，test data 的 accuracy、precision、recall、f1-score 皆為 99.37%，相當精準。

CNN 模型對於每一張圖片，會對每一個分類提出一個預測值，我們取最大的那一個值所屬的分類作為預測結果，下圖是拿其中一張圖片來測試，這張圖片是一顆新鮮的蘋果，resize 成 128*128*3 之後，給分類器判別，對於六個類別分別提出預測值，其中最大的是第 0 類（新鮮的蘋果），預測值為 24.5091，正確預測類別。




## 實驗二

為了跟後面的 SVM 模型比較，希望輸入的影像大小同為 64*64*3，因此我們再訓練了一個 CNN 模型，輸入大小為 64*64*3，其他參數只略微修改，和輸入符合。右圖比較兩個 CNN 模型的結果，包含 train data 的 accuracy，及 test data 的 accuracy、precision、recall、f1-score，兩個模型的結果接近，輸入大小為 64*64*3 的模型的預測結果略高一點點。



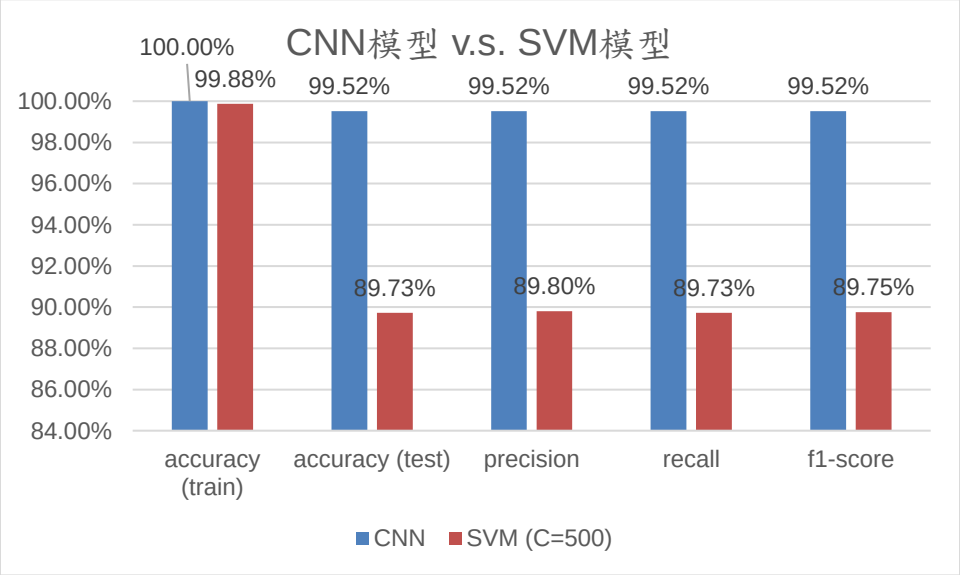
## 實驗三

我們建立 SVM 模型，並比較不同參數。一開始使用 sklearn library 的 svm 建立模型，但執行時間過長。後來改用 thundersvm 的 svc 建立模型。原本希望把影像 resize 成 128*128*3，但在 Google Colab 執行時 RAM 不足，改成 64*64*3。我們比較了不同的參數 C 對準確度的影響。C 是對誤差的寬容度，C 越高代表越不能有誤差，容易出現 overfitting。C 太小則可能準確度不佳。下表為我們的實驗結果。

| C | 10 | 100 | 500 | 1000 |
|---|---|---|---|---|
| accuracy (train data) | 87.17% | 96.69% | 99.88% | 99.99% |
| accuracy (test data) | 83.73% | 89.44% | 89.73% | 89.55% |
| precision (weighted avg) | 83.86% | 89.75% | 89.80% | 89.61% |
| recall (weighted avg) | 83.73% | 89.44% | 89.73% | 89.55% |
| f1-score (weighted avg) | 83.64% | 89.47% | 89.75% | 89.56% |

實驗結果可以看到，當 C=500 時 test data 的 accuracy、precision、recall、f1-score 都最高，但當 C 增加到 1000 時，train data 的 accuracy 略為上升，只有一張圖片判別錯誤，但 test data 的 accuracy 略為下降，可能是 overfitting 的情況。因此當 C 為 500 時，是這幾個模型中最適合 kaggle 這個資料集的。

接著比較了 CNN 模型和 SVM 模型的結果，CNN 模型選擇影像大小為 64*64*3 的那一個，SVM 模型前面訓練的影像皆用 64*64*3，因此選擇表現較優的那一個模型（C=500）。訓練時間 CNN 模型 2111.14 秒，SVM 模型 64.09 秒，訓練 CNN 模型花費的時間比 SVM 多，但各項評估指標的值高出 SVM 模型約 10%，表現較佳。




## 實驗四

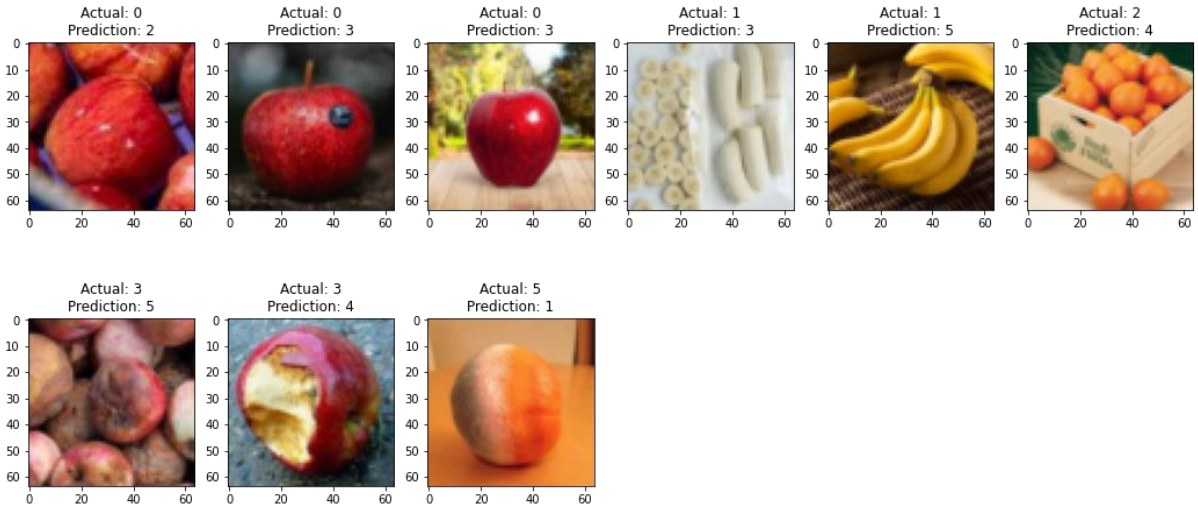
我們將爬蟲抓取下來整理過的圖片 resize 成 64*64*3，給我們在實驗二、三訓練出來的分類器判別。以下為實驗結果。

| 模型 | CNN | SVM (C=10) | SVM (C=100) | SVM (C=500) | SVM (C=1000) |
|---|---|---|---|---|---|
| accuracy (google test data) | 85.48% | 56.45% | 48.38% | 46.77% | 48.38% |
| precision (weighted avg) | 86.80% | 61.73% | 53.66% | 49.92% | 51.71% |
| recall (weighted avg) | 85.48% | 56.45% | 48.39% | 46.77% | 48.39% |
| f1-score (weighted avg) | 85.55% | 57.15% | 48.61% | 46.78% | 48.60% |

觀察數據可以發現，雖然實驗二、三時，CNN 跟 SVM 的分類器準確率都可以達到八成以上，但當測試資料改成我們從網路上抓取的照片時，SVM 的準確率巨幅下滑，有三組都小於五成，並且在實驗二、三中表現越好的 SVM 分類器，在此實驗中的表現越差（除了 C=1000 的）。相較之下，CNN 準確率仍有 85%。

仔細觀察 kaggle 的圖片，可以發現圖片雖有經過旋轉、平移等方式處理，但每張圖片裡物體的大小相去不遠，且均為黑底或白底。而實驗四使用的資料亮暗程度不一、背景有些複雜有些單一、物體大小也不一。

我們觀察 CNN 預測的結果，各類別的正確率分別為 8/11, 13/15, 8/9, 6/8, 10/10, 9/10，其中白底的圖片都被預測正確了，預測錯誤的都是背景複雜的圖片。單就背景複雜（非白底或黑底）的圖片而言，被誤判的比率約為三成。此外，新鮮的較容易被誤認為腐爛的。而物體的大小似乎沒有造成太大影響，比較小的新鮮橘子跟腐爛橘子等，都有被預測正確。下圖為 CNN 預測錯誤的圖片。




再觀察 SVM 預測的結果，各類別的正確率分別為 6/11, 8/15, 5/9, 4/8, 8/10, 5/10，蘋果和橘子的正確率都只有五成左右，而香蕉的準確率相較之下稍微高一些。且預測錯誤的蘋果多半是被判斷成橘子。Zhang 等人的研究[3]提到使用背景過度單一的圖片作為訓練資料集時，在複雜的測試資料集上會表現不佳。由於我們是使用背景單一的圖片做訓練，這或許是造成我們 SVM 分類器結果不佳的原因。

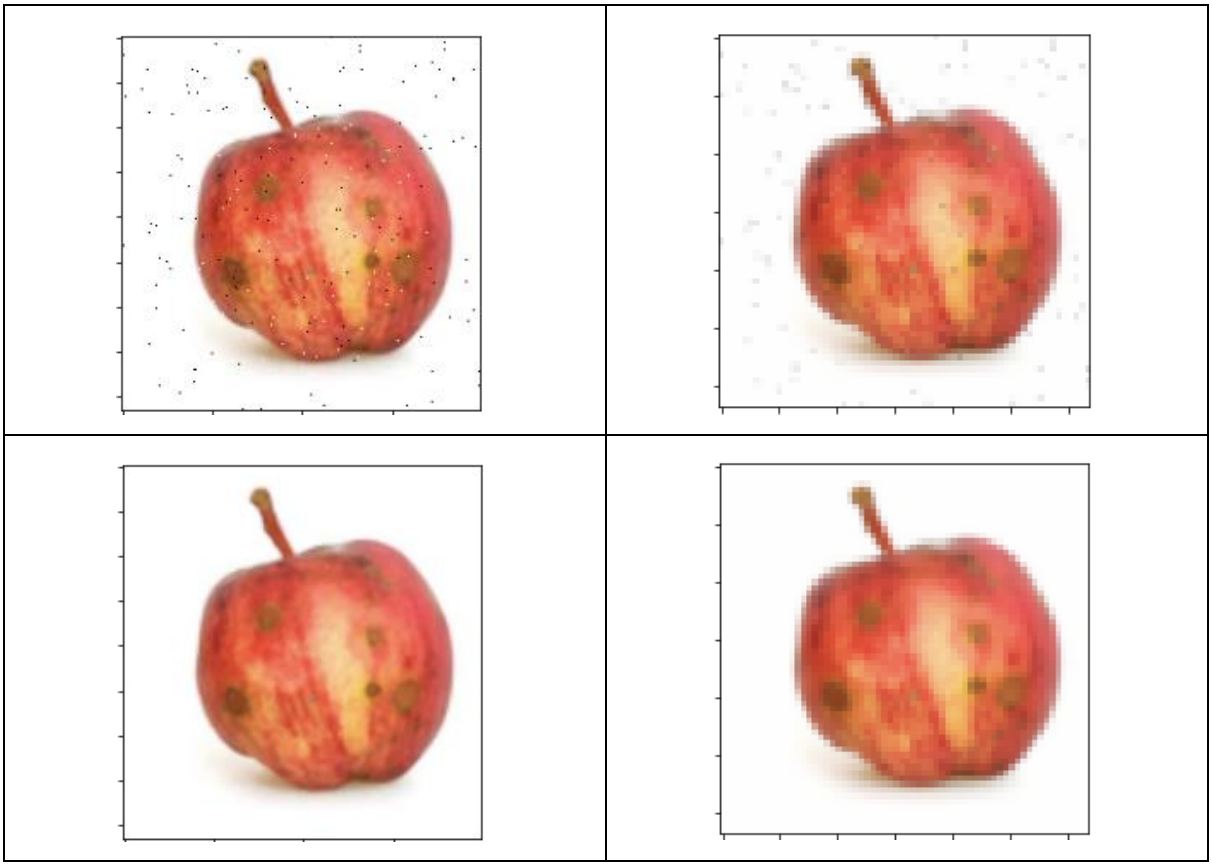
在以上兩種分類器的預測裡，香蕉的準確率都是最高的，而蘋果則是最低。

## 實驗五

這個實驗我們想測試椒鹽雜訊對模型的影響，原本 Kaggle 的資料集中部分圖片已加入椒鹽雜訊，圖片名稱以「saltandpepper」為開頭，下表是各個類別含有椒鹽雜訊的數量。

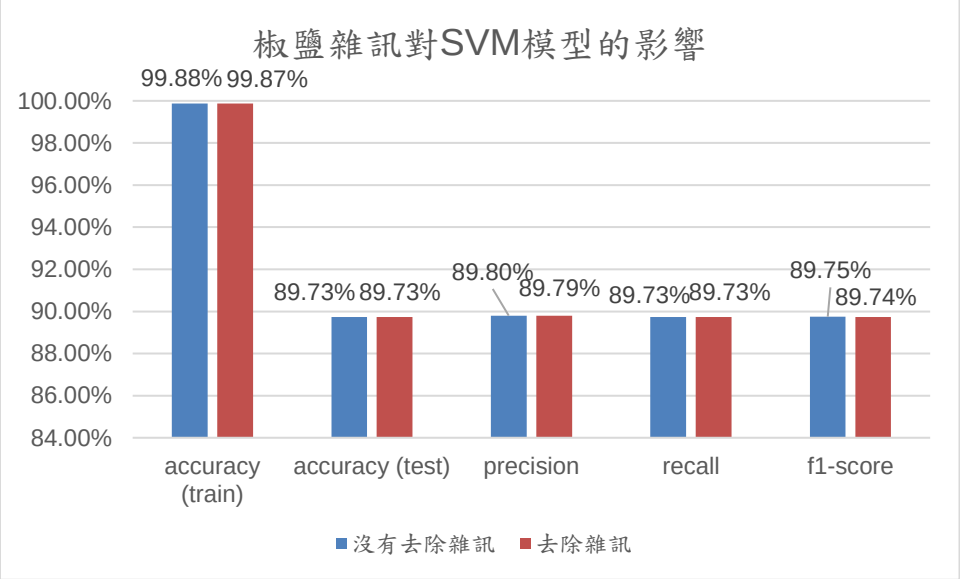
| | train data | test data |
|---|---|---|
| fresh apples | 187 | 45 |
| fresh banana | 180 | 38 |
| fresh oranges | 161 | 45 |
| rotten apples | 260 | 67 |
| rotten banana | 243 | 63 |
| rotten oranges | 174 | 48 |
| 椒鹽圖片總計 | 1205 | 306 |
| 所有圖片總計 | 10901 | 2698 |

對於這些加入了椒鹽雜訊的圖片，我們用 3*3 的 median filter 過濾。Median filter 是對區域內留下中間值，因為椒鹽雜訊是白點（最大值）和黑點（最小值），用 median filter 可以濾除。雖然會造成圖片稍微模糊了一點，但在 resize 成 64*64*3 之後影響不大。以下是以其中一張圖片作為示範。左上、右上、左下、右下依序是(a) 原圖 (b) 把 a resize 成 64*64*3 (c) 用 median filter 處理 a (d) 把 c resize。從圖中可以看到，經過處理的圖 d 比圖 b 少了雜訊造成的模糊的點。



首先我們拿之前訓練好的模型，對 test data 中含有椒鹽雜訊的圖片，比較去掉雜訊前後的預測結果。因為 CNN 模型的準確度已經非常高，模型可能已經自動對於雜訊有所取捨，我們選擇用 SVM 模型（C=500）來測試。對於含有椒鹽雜訊的 306 張圖片，在去掉雜訊前，accuracy 為 94.12%，去掉雜訊後的 accuracy 為 94.44%。詳細比較了預測結果，只有兩張圖片預測結果不同，兩張都是腐爛的蘋果且原本預測為腐爛的橘子，去掉雜訊後，一張預測正確，另一張改預測成新鮮的蘋果。因此我們推測去掉雜訊可能對於預測稍微有一點影響，但影響不大。

我們把 train data 中含有椒鹽雜訊的圖片去掉雜訊，重新訓練 SVM 模型，下表是比較新舊模型的結果。從實驗來看，兩者在評估指標的值幾乎沒有差別。




## Conclusion

我們用 CNN 和 SVM 兩個演算法產生的分類器判別新鮮及腐爛的蘋果、香蕉、橘子，對於 Kaggle 上的 Fruits fresh and rotten for classification 這個資料集，CNN 模型的準確度為 99.52%，SVM 模型的準確度為 89.73%，CNN 模型的表現較為優異。可能因為 Kaggle 的資料集較為乾淨，圖片經過比較好的整理，並且只有六類，類別較少，且類別之間較容易分辨，準確度較高。使用 Google 圖片作為測試集後，我們發現 CNN 模型即便只有使用單一背景的圖片訓練，在單一、複雜背景混合的測試集上都有不錯的表現。相較之下，使用單一背景的圖片訓練的 SVM 分類器在這種測試集上的表現欠佳。我們也比較了椒鹽雜訊對分類器的影響，實驗顯示幾乎不影響判別。

## References


1. Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng and M. Chen, "Medical image classification with convolutional neural network," 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), 2014, pp. 844-848, doi: 10.1109/ICARCV.2014.7064414.

2. D. Lu & Q. Weng (2007) A survey of image classification methods and techniques for improving classification performance, International Journal of Remote Sensing, 28:5, 823-870, DOI: 10.1080/01431160600746456
3. Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, Cordelia Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. International Journal of Computer Vision, Springer Verlag, 2007, 73 (2), pp.213-238. ff10.1007/s11263-006-9794-4ff. ffhal-00171412
4. Noise or Signal: The Role of Backgrounds in Image Classification
5. 資料集來源：https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification
6. 深度學習：CNN 原理. 想必剛踏入深度學習 Computer… | by Cinnamon AI Taiwan
7. ML 2021 Spring
8. Next Image Classification Techniques in Remote Sensing [Infographic]
9. Image Classification Using Machine Learning-Support Vector Machine(SVM)
10. Batch Normalization 介紹. 隨著神經網路越來越深，為了使模型更加穩定，Batch… | by 李謦伊| 謦伊的閱讀筆記
11. 线性整流函数
12. 卷积神经网络
13. 【机器学习理论】分类问题中常用的性能评估指标
14. 機器學習-支撐向量機(support vector machine, SVM)詳細推導
15. 機器學習: Kernel 函數. 在機器學習內，一般說到 kernel 函數都是在 SVM 中去介紹，主要原因是 SVM 必須…
16. [ML] 機器學習技法：第三講 Kernel SVM