Report

Statement

小時候放學回家，最期待的就是飯前收看電視播映的卡通，蠟筆小新、哆啦A夢...佔據了我的童年。我們知道，每位畫家的作品都有獨自的繪畫風格，因此，我不禁好奇電腦是否也能分辨出這些卡通的差異呢?

這次的實作我從Youtube上蒐集5部不同卡通的擷取畫面，訓練3種模型，測試並分析它們分類卡通的能力。

當我在Kaggle上搜尋類似的卡通分類主題時，看到有位前人也是用差不多的做法建立資料集，而他的樣本是以美國卡通為主，且不侷限於"人類"角色；此外，他的截圖頻率比較高，所以有許多圖片看起來是重複的。在我的實作中改善了上述缺點。

Dataset

- Data Type:
  .jpg檔

- Amount and Composition:
  共5部日本卡通，每部卡通各包含520張圖片(1088 x 1920):
  蠟筆小新、哆啦A夢、我們這一家、花田少年史、櫻桃小丸子。
  儲存這些圖片的資料夾名稱即為他們的標籤。

- Conditions:
  挑選的5部卡通都是以"人類"角色為主，且這些人物的共同特色是臉部線條都屬於"圓滑"的類型(相對於動漫人物有稜有角的風格而言)。

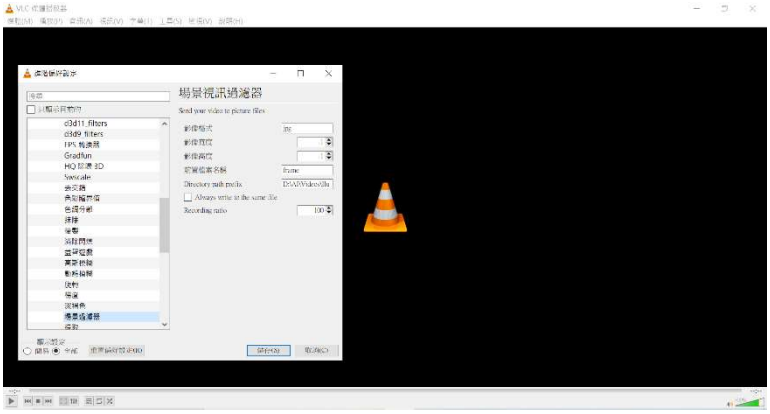
  每部卡通的圖片來源至少取自"5集"，且故事內容經過挑選，確保場景、人物等重複出現次數不會太高(否則模型可能只是學到某些人物的特徵而已)。例如其中一集是小丸子和家人一起聊天，那另一集就會選在學校和同學玩耍的場景。此外，取樣時以每100(或200)幀截圖一次的頻率，同樣也是避免間隔太近的畫面相似度太高。

  圖片品質的部分，剔除畫面邊緣有太多商標或跑馬燈的樣本，此外，模糊(例: 幀和幀切換的瞬間)的畫面也會去掉。

- Process:
  1. 在Youtube上挑選適合的卡通集數，以全螢幕撥放並螢幕錄影。
  2. 使用"VLC media player"，以每100/200幀取樣的頻率，將錄製好的

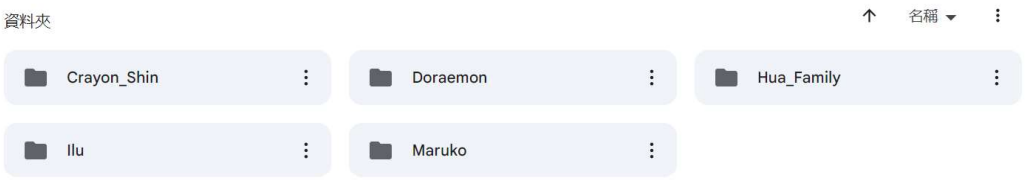
影片擷取為多張圖片。

3. 人工檢查，確保圖片品質。



VLC操作畫面與取樣設定

- Examples:



5部卡通，資料夾名稱即為標籤

Ilu資料夾下的圖片

Doraemon資料夾下的圖片

- **Methods**
  - SVM (supervised)

Libraries:

sklearn - train_test_split, cross_validate, SVC, accuracy_score, confusion_matrix

Hyperparameters:

- re_size = (196, 196) / (144, 260)
- svc_c = 5.0
- k = 3
- num_classes = 5

步驟:

I. 圖片預處理。包含將圖片的色彩通道轉換至RGB、統一圖片大小、展開為1D數組、圖片數值正規化。
II. 準備資料集。將所有圖片以8:2的比例切分為訓練集與測試集。
III. 定義SVC模型。C值為5.0；kernel、gamma等其他參數採預設值。
IV. 用訓練集進行K折交叉驗證訓練模型。
V. 選擇K折中表現最佳的模型，再用整個訓練集對其訓練一次。
VI. 在測試集上做分類，以評估訓練後的模型效能。

Opensource Code:

SVM Model

■ CNN (supervised & DL)

Libraries:

- *torch* - DataLoader, random_split, Subset, Adam, CrossEntropyLoss
- *torchvision* - transforms, resnet18
- *sklearn* - KFold, accuracy_score, confusion_matrix
- *PIL* - Image

Hyperparameters:

- re_size = (196, 196) / (144, 260)
- batch_size = 32
- lr = 0.001
- k = 3
- num_classes = 5
- epochs = 2

步驟:

I. 定義圖片預處理。先將色彩通道轉換至RGB，再經由transforms的操作統一圖片大小、隨機水平翻轉、正規化。
II. 準備資料集。將所有圖片以8:2的比例切分為訓練集與測試集。
III. 用訓練集進行K折交叉驗證訓練模型，在每一折中:

i. 使用在ImageNet上預訓練的ResNet18模型，並將fully connected layer的輸出維度改為5(class數量)。
ii. 將模型移動至GPU。
iii. 設定optimizer和loss function。
iv. 開始訓練模型。執行2 epochs，過程利用梯度和損失更新參數。
v. 每epoch訓練結束後，使用該折的測試子集評估模型效能。

IV. 選擇K折中表現最佳的模型，再用整個訓練集對其訓練一次。
V. 在測試集上做分類，以評估訓練後的模型效能。

Pretrained Model:

- ResNet18 pretrained on ImageNet - *resnet18(pretrained=True)*

Opensource Code:

Apply K-Fold Cross Validation on ResNet

■ K-means Clustering (unsupervised)

Libraries:

- *sklearn* - KMeans, accuracy_score, adjusted_rand_score

Hyperparameters:

- re_size = (196, 196) / (144, 260)
- num_classes = 5

步驟:

I. 圖片預處理。包含將圖片的色彩通道轉換至RGB、統一圖片大小、展開為1D數組、圖片數值正規化。
II. 定義K-means模型。分為5個聚類，並設定進行10次以不同初始中心的訓練。
III. 用整個資料集訓練模型，得到各個聚類中心以及每張圖片的預測聚類標籤。
IV. 使用預測聚類標籤評估訓練後的模型效能。

Opensource Code:

K-means Clustering

■ PCA (dimensionality reduction)

Libraries:

sklearn - PCA

Hyperparameters:

n_components = 300

步驟:

I. 定義PCA模型，設定降維至300D。
II. 用資料集訓練模型。

Experiments

共進行4種實驗:

	image size	PCA	amount of data
Exp1.	196 x 196	x	1
Exp2.	196 x 196	o	1
Exp3.	144 x 260	x	1
Exp4.	196 x 196	x	1/2

■ Description:

Exp1.是最基本的實驗設定。將圖片邊長縮放至1:1，尺寸為196是受限於硬體容量限制而定，不使用降維技術，且使用完整的資料集(共2600張圖片)。

Exp2.所有參數設定與Exp1.相同，但嘗試使用了降維的技術，將圖片映射到300D的空間中再進行訓練。對於此實驗，我的預期是模型表現會差不多，因為所使用的PCA技術是在盡量保持原特徵的情況下，對高維度的資料進行降維，所以結果不會有太大的差異；而降維不免會失去一些細節，但只要維度不降太多，保持在足夠代表特徵的限度內，對模型表現的影響應該不會太大。

Exp3.也是和Exp1.大致相同，但是讓圖片縮放後的長寬比例與原圖盡量一致(長邊大概是寬邊的1.8倍)，且像素總數也和其他實驗組差不多(38416 v.s.37440)。我的預期是模型表現會略好一點，因為原本的圖片是長方形，當縮放成正方形時，長邊被壓縮得比較多，可能會失去一些訊息，因此讓圖片等比例縮放應該會比較好。

Exp4.則是參考了Spec.的建議，僅使用一半的資料集訓練模型。當訓練資料較少時，神經網路和無監督的方法理論上結果會變差；而SVM在上課時有提到，它在小樣本上訓練的表現通常會比其他模型好，但我不太確定我的1/2樣本會不會太少，反而使模型underfitting，因此對這部分的實驗結果我比較沒有把握。

1. SVM

■ Evaluation Results:

(Accuracy Confusion Matrix)

Exp1. 0.7769

$$\begin{bmatrix} 78 & 7 & 3 & 10 & 6 \\ 13 & 82 & 3 & 9 & 3 \\ 6 & 1 & 97 & 4 & 2 \\ 10 & 1 & 0 & 86 & 7 \\ 8 & 10 & 2 & 11 & 61 \end{bmatrix}$$

Exp2. 0.7904

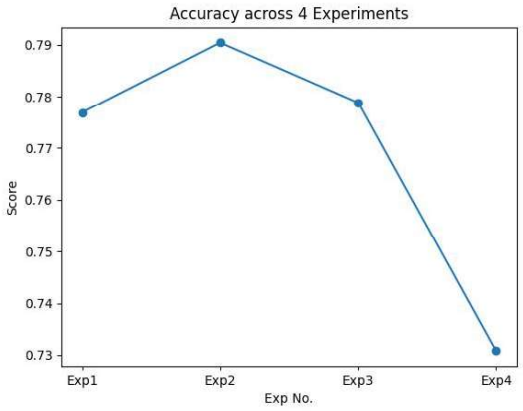
$$\begin{bmatrix} 77 & 10 & 2 & 8 & 7 \\ 13 & 85 & 1 & 7 & 4 \\ 8 & 1 & 99 & 0 & 2 \\ 9 & 1 & 0 & 86 & 8 \\ 9 & 10 & 2 & 7 & 64 \end{bmatrix}$$

Exp3. 0.7788

$$\begin{bmatrix} 77 & 8 & 3 & 10 & 6 \\ 12 & 84 & 3 & 9 & 2 \\ 7 & 1 & 96 & 4 & 2 \\ 8 & 1 & 0 & 87 & 8 \\ 9 & 8 & 2 & 12 & 61 \end{bmatrix}$$

Exp4. 0.7308

$$\begin{bmatrix} 37 & 6 & 2 & 4 & 3 \\ 7 & 33 & 2 & 5 & 6 \\ 4 & 1 & 49 & 1 & 1 \\ 2 & 4 & 0 & 46 & 4 \\ 5 & 6 & 2 & 5 & 25 \end{bmatrix}$$

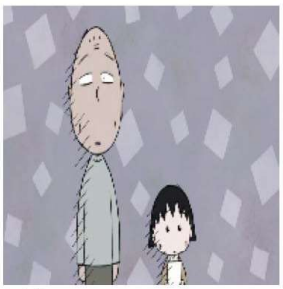


比較四組實驗的正確率可以觀察到，使用 PCA 降維後的資料集表現最佳；等比例縮小的方式確實有略好一點；而使用一半數量的資料集，結果差比較多。Exp2 降維後的結果變好，可能是因為降維的過程消除了部分噪音，並選擇了更具代表性的特徵，因此超平面能更有效地將不同類別分開，使正確率提升；而 Ex3、Ex4 的結果與我預期的差不多。

觀察混淆矩陣可以發現，「我們這一家」的 Recall 及 Precision 都是最高的，可知其最具有辨識度；而「櫻桃小丸子」最常被預測錯誤，且容易與「哆啦 A 夢」、「花田少年史」搞混。另外我也發現，在「哆啦 A 夢」的 FN 中，其最常被誤認為是「蠟筆小新」。

■ Examples:

Hua_Family的正確率最高



Maruko預測為Doraemon　　*Doraemon預測為Craypn_Shin*

2. CNN

■ Evaluation Results:

(Accuracy　　Confusion Matrix)

Exp1. 0.9038

$$\begin{bmatrix} 96 & 0 & 1 & 1 & 1 \\ 15 & 87 & 1 & 1 & 5 \\ 2 & 0 & 96 & 0 & 2 \\ 0 & 0 & 3 & 113 & 0 \\ 15 & 0 & 3 & 0 & 78 \end{bmatrix}$$

Exp2. 0.4038

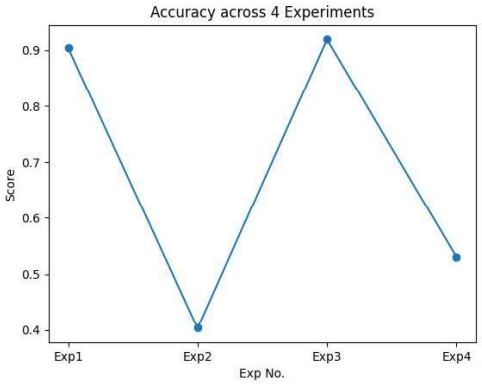
$$\begin{bmatrix} 77 & 2 & 15 & 18 & 0 \\ 68 & 1 & 23 & 15 & 4 \\ 19 & 0 & 65 & 10 & 0 \\ 31 & 0 & 7 & 56 & 3 \\ 48 & 0 & 31 & 16 & 11 \end{bmatrix}$$

Exp3. 0.9192

$$\begin{bmatrix} 93 & 0 & 1 & 0 & 14 \\ 3 & 89 & 0 & 0 & 5 \\ 3 & 0 & 94 & 0 & 4 \\ 4 & 6 & 0 & 86 & 2 \\ 0 & 0 & 0 & 0 & 116 \end{bmatrix}$$

Exp4. 0.5308

$$\begin{bmatrix} 20 & 0 & 0 & 41 & 4 \\ 0 & 42 & 0 & 8 & 0 \\ 0 & 0 & 8 & 43 & 0 \\ 0 & 1 & 0 & 50 & 0 \\ 2 & 10 & 0 & 13 & 18 \end{bmatrix}$$



CNN 這四組實驗的變化趨勢和其他兩者差異甚大。可以看到，等比例縮小的方式表現最佳，符合我的預期；但使用 PCA 降維與一半資料集的結果卻大幅降低，尤其是降維後的模型，正確率不到原始的一半。

先分析 Ex2.的原因，我覺得 CNN 在這個情況下表現極差，與模型對輸

入形狀的要求有關。其他兩個模型可以直接以降維後的 1D 陣列作為輸入；但 CNN 要求輸入的形狀必需是(C, H, W)，因此，我將降維後的 1D 陣列 reshape 回(3, H, W)的格式時(由於使用預訓練的權重，所以需要是 C=3)，每個數值可能已經不代表原來的位置，導致 CNN 在學習像素之間的關係特徵時就遇到了問題。

至於 Ex4，在所有模型中這個實驗的表現都會變差，但在這裡正確率下降的程度特別明顯。我覺得是因為神經網路的參數量很多，所以對訓練資料的數量特別敏感。使用一半的資料集顯然數量不足，導致模型 underfitting。

觀察混淆矩陣，Ex2 的模型似乎真的學錯了特徵，很多情況都猜成了「蠟筆小新」；Ex4 的矩陣看起來也有異常，大部分的猜測都是「花田少年史」。

- Examples:

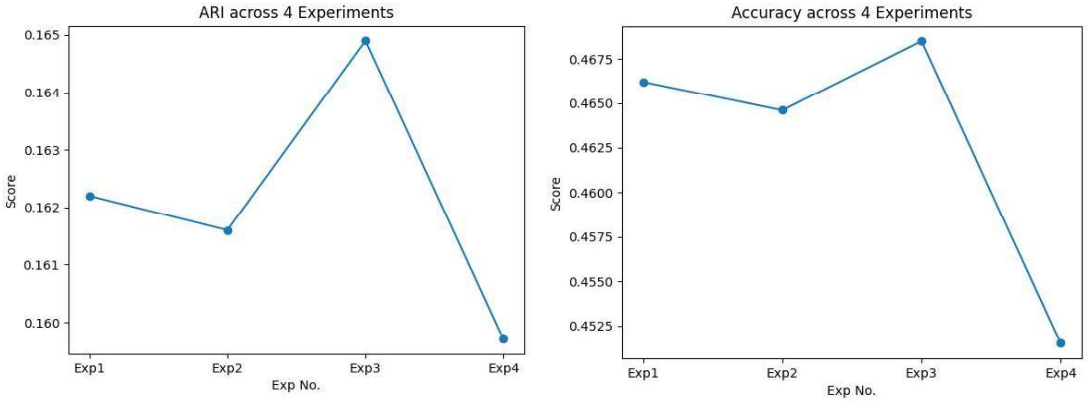
Exp2.許多預測都是Craypn_Shin

Exp4.許多預測都是Ilu

3. K-means Clustering

- Evaluation Results:

	(ARI,	Accuracy)
Exp1.	(0.1622,	0.4662)
Exp2.	(0.1616,	0.4646)
Exp3.	(0.1649,	0.4685)
Exp4.	(0.1597,	0.4515)



從這兩個指標可以觀察到共同的趨勢，即等比例縮小的表現最佳；PCA降維後略差一點；而一半資料集則有較大的下降。Ex3和Ex4的結果同樣如我預期；而至於Ex2中經過降維處理後，為何對SVM有幫助但在這裡卻相反呢?我的推論是，由於K-means Clustering是非監督式學習的方法，因此多一點的特徵對於它學習underlying information可能越有幫助，所以在這裡執行降維可能較沒那麼有益。

而整體來說，K-means Clustering的表現和其他監督式學習的方法相比差很多，因為它是在沒有真實標籤的情況下，透過自行學習資料特徵來訓練的，所以模型表現上會有所限制。

■ Examples:

Clustering的結果似乎較不穩定

● Discussion

1. Based on your experiments, are the results and observed behaviors what you expect?

我原先的預期是CNN的表現會最好，其次是SVM，最後是K-means Clustering。因為CNN最初就是為圖片任務而設計的神經網路，而且它是三者中唯一有保留圖片空間訊息的模型 (SVM和K-means的輸入都必須是1D數組)，其次，CNN是深層的神經網路，並且我利用了在ImageNet上預訓練的模型，所以它的性能相對要比較好；Clustering屬於非監督式的訓練，所以預期其學習效果會最差；而SVM就是介於兩者之間。

SVM在各實驗的結果和我預期的最接近；CNN在Ex2中大幅下滑的結果令我非常意外，這是因為我一開始忽略了降維後會打亂位置訊息的問題。另外，雖

然知道Ex4的表現會下降，但實際下降的比例也比我預期的多；而Clustering在Ex2中略微下降的表現則是我比較沒有預料到的。以上可能造成的原因皆已在Experiments - Evaluation Results中分析。

2. Discuss factors that affect the performance, including dataset characteristics.

根據我的實驗結果，資料維度、縮放大小、訓練資料數量都會影響模型的表現。此外，我這次沒有進行額外的特徵提取，如果加入這個處理步驟，模型的表現應該會更好。另外，還有其他影響神經網路的因素，例如batch size, learning rate, epochs等。

資料集本身的屬性也會有影響，例如資料分布、特徵的相關性、雜訊等。在我的資料集中，每個類別的數量都一樣，所以沒有資料不平衡的問題。

3. Describe experiments that you would do if there were more time available.

如果有時間的話，我想嘗試只擷取卡通人物的”臉部”來進行分類。因為畫風的差異主要在於人物特徵，所以我想測試電腦是否能在不受背景等其他因素的影響下，僅單純觀察人物風格就能正確分辨出不同卡通的差別。

如果要實作這部分的話，可能需要用到一些臉部檢測的技術，如OpenCV或Dlib。

4. Indicate what you have learned from the experiments and remaining questions.

蒐集資料集的部分，在選擇主題以及搜尋資源的過程中我有一些心得: 想要建立一個合理、有用的資料集需要考慮很多面向。我決定做卡通的分類，接著我要將這個問題做更嚴謹的定義，包含侷限在”人物”角色、各類別的風格挑選(例如我原本還準備了「葬送的芙莉蓮」的影片，但思考後認為卡通與動漫的風格、線條等差異太大，不適合放在一起分類)，以及為了避免圖片過於相似，我也調整了截圖的頻率。如此，使得問題具有挑戰性且有意義。

在技術方面，這是我第一次練習使用無監督學習的方法，雖然效果不如監督式的模型好，但不須標籤的訓練方式非常有趣!我也順便學到了一些不需要真實標籤的評估指標。另一個嘗試是使用 PCA 降維方法，雖然它對於各個模型帶來的影響不同，但明顯感受到的是計算效率提升了(訓練過程跑得超快的)!

● References

Cartoon Classification

Cross-Validation

K-means Clustering 介紹

PCA 介紹

Adjusted Rand Index