

Deep Learning 2024 HW1

Image Classification with Multiple Models and Features

資訊所 P76124752 莊上緣

Github: [shangyuan191/DL2024_RE50800_HW1 \(github.com\)](https://github.com/shangyuan191/DL2024_RE50800_HW1)

1. 題目介紹

在近年的 AI 領域中，影像相關的主題一直是主流研究方向之一，此次作業要求使用多個模型與特徵來進行圖像分類，最終希望能解決圖像分類問題，並通過結合多個模型和特徵來提高預測表現與分類性能。

2. 資料集介紹

給定 TinyImageNet 資料集，此資料集內共有 200 個種類的照片，每個種類分別有 500 張照片，因此總共是 100000 張照片。

3. 前處理

(1) Reshape:

在 TIN 資料集內，每張照片的預設 shape 為 (64, 64, 3)，由於之前做過 cifar10 資料集影像分類相關的作業，因此我將本次作業的圖片都重新 reshape 成與 cifar10 圖片相同的 (32, 32, 3)，也可以在一定程度上加快模型訓練的時間(但相對地可供提取的特徵就會變少)

(2) Train_test_split:

我以 4:1 的比例將資料集進行切分，80%的資料給模型進行訓練，20%的資料用來進行 test。

(3) 部分抽樣:

由於原本的資料集 label 有 200 種，相當龐大，因此我只從 200 類圖片中選出 100 類進行圖片分類。結合 4:1 的 train test 比例，我選擇 40000 張照片進行 training，10000 張照片進行 testing

4. 特徵提取方法

(1) HOG:

HOG(方向梯度直方圖)專門用來檢測圖像中的邊緣和紋理，首先將圖像分成很多個小 unit(aka cell)，計算每個 cell 中的梯度方向，再將這些梯度方向組成直方圖，最後再進行正規化。

簡而言之，HOG 能幫我們把圖像中的邊緣和紋理轉成數字表示並提取出來，讓模型能更好地了解與處理圖像。

(2) ColorHistogram:

這個特徵提取方式是這三個方式中唯一不需要進行灰階處理的，而是統計 RGB 的 pixel 數量，並將這些統計結果組成一個直方圖，每個 channel 的直方圖代表該圖像中的 color channel 的分布情況，亦即不同亮度和顏色的 pixel 出現的頻率。

(3) **SIFT:**

SIFT 是一種 CV 領域的演算法，用來偵測與描述影像中的局部性特徵，在空間尺度中的極值點，並提取該極值點的位置、尺度、旋轉不變數

5. 模型

(1) **KNN:**

K-近鄰演算法為一個經典的分類算法，其原理就是將資料點分布的座標空間中，對於每個未知 label 的點，找尋與他歐式距離最近的 K 個有 label 的點，進行投票決定該未知點的 label 為何。

(2) **SVM:**

SVM 是一個用於分類&迴歸的監督式學習模型，核心概念是在特徵空間中找到一個最佳的 hyperplane，這個 hyperplane 可以視為一個決策邊界，用來將兩個類別的樣本分開，因此使用多個 hyperplane 就可以處理多類別的問題，其優點是有效處理高維空間數據、kernel function 的選擇也相對靈活，但缺點是訓練時間較長(詳見實驗數據表格)

(3) **RandomForest:**

這是一種 ensemble 的機器學習方法，由多棵 decision tree 構成隨機森林模型，每棵決策樹都是獨立訓練，最終通過投票或取平均的方式來進行預測。其一大特性是訓練過程中會隨機抽樣選擇部分樣本與一部份特徵，因此可以增加每棵樹的多樣性，並有效減低 overfitting 問題，也很適用於大規模資料集

6. 實驗結果

Hyper-parameter 組合(一):

KNN:K=5

SVM:kernel=linear, C=1.0, gamma=auto

RandomForest:n_estimator=50

(1) **HOG:**

	Accuracy	F1 score	Run time(sec)
KNN	0.09	0.08	12.96
SVM	0.13	0.11	470.39
RandomForest	0.08	0.07	175.56

(2) ColorHistogram:

	Accuracy	F1 score	Run time(sec)
KNN	0.07	0.07	6.18
SVM	0.12	0.10	556.83
RandomForest	0.14	0.12	49.98

(3) SIFT:

	Accuracy	F1 score	Run time(sec)
KNN	0.009	0.0001	171.67
SVM	0.01	0.0001	339.60
RandomForest	0.01	0.0001	171.96

Hyper-parameter 組合(二):

KNN:K=20

SVM:kernel=rbf, C=1.0, gamma=auto

RandomForest:n_estimator=50

(1) HOG:

	Accuracy	F1 score	Run time(sec)
KNN	0.1	0.09	7.02
SVM	0.08	0.06	535.59
RandomForest	0.08	0.07	168.83

(2) ColorHistogram:

	Accuracy	F1 score	Run time(sec)
KNN	0.09	0.08	6.50
SVM	0.09	0.07	850.99
RandomForest	0.14	0.12	49.63

(3) SIFT:

	Accuracy	F1 score	Run time(sec)
KNN	0.01	0.0002	171.72
SVM	0.01	0.0001	397.00
RandomForest	0.01	0.0001	171.98

實驗觀察：

1. KNN 由於算法較為簡單，因此 run time 也都是最快的
2. 在將 KNN 的 K 值由 5 調至 20 後，發現 accuracy 與 F1 score 都有微幅上升，推測可能是因為 label 有 100 種，但原本的 k 只有 5 個，可能沒辦法進行較為準確的近鄰投票，還是要「參考多方意見」
3. 經實驗發現，SVM 的 kernel 選 rbf，相較於選 linear 而言，不僅 run time 更長，最終的表現也更差，推測可能是因為高斯核函數計算相對複雜，出現 overfitting 的狀況(或是搭配的 C 與 gamma 值所導致)
4. 總體而言，**表現最好的組合為 HOG+SVM 與 ColorHistogram+RandomForest**

7. 結論與心得

我認為如果算力允許或時間成本允許的話，或許可以嘗試用 grid search 來暴力找出最佳參數組合，或是使用其他特徵萃取方式、其他模型來進行測試。

在我過去的實作經驗中，ML-based 的方法通常是用來處理維度、特徵向量長度、資料量皆沒這麼大的表格式資料，影像相關的問題則通常是使用深度學習的方式去處理，而這次作業是我第一次使用 ML 方法來處理影像分類的問題。顯而易見地，就算搭配上特徵萃取方法，這些模型表現的結果仍然並不佳，這說明了傳統機器學習的方式在影像分類問題上的受限狀況，因此未來或許可以朝著深度學習方法應用於影像分類任務的方向去做嘗試，透過多層神經網路的特徵萃取，有望提高面對這種影像辨識分類任務時的預測表現。

8. Reference

[Introducing ChatGPT \(openai.com\)](https://openai.com/)

[Object Detection — HOG + SVM. 傳統最強大的物體偵測方法 | by Jeremy Pai | Life's a Struggle | Medium](#)

[一文讲解方向梯度直方图（hog） - 知乎 \(zhihu.com\)](#)

[5.4. SIFT\(Scale-Invariant Feature Transform\)簡介 - OpenCV Python Tutorials \(opencv-python-tutorials.readthedocs.io\)](#)

[SIFT 算法详解-CSDN 博客](#)

[ML 入門（十七）隨機森林\(Random Forest\). 介紹 | by Chung-Yi | 程式設計之旅 | Medium](#)

[看了这篇文章你还不不懂 SVM 你就来打我 - 知乎 \(zhihu.com\)](#)