

# Deep Learning HW1

RE6111024 葉嘉宏

## Abstract

本次作業為利用影像分類的任務，複習機器學習的概念。讀取影像後，利用幾個常見的特徵計算方式，取得各種影像的特徵，並使用常見的機器學習模型，例如 XGBoost、LightGBM..... 等，對影像進行分類。程式碼的 github 在 <https://github.com/chyeh1126/DeepLearning-2023>。

## I. Introduction

在電腦視覺領域中，常見的任務包含影像分類、物件偵測、..... 等。影像分類的目的是藉由影像呈現的特徵，讓模型輸出影像中包含某些物件的機率值，以此判斷影像中是否存在某些物件。

## II. Methodology

在本次的影像分類任務中，由於每張影像的大小不盡相同，且一張影像的資料量較大，需要先提取特徵後才能輸入模型做學習，同時也會降低訓練所需要的時間。由於本次作業使用的模型都是常見的機器學習模型，這裡不會特別介紹各個模型，會著重在介紹有使用的特徵計算方式。

### A. Global Color Histogram

影像中的每個像素都有特定的 R、G、B 數值，可將三個元素繪製成直方圖，檢視不同色彩在影像中的分布，此即 Global Color Histogram 的核心概念。此計算方式主要考量色彩佔比，而非色彩在影像中的位置，可以用於描述較難被自動分割的影像。此外，同一張影像經過旋轉或是縮放，繪製出的 Color Histogram 不變。

此特徵也隱含一些缺點。因為不考慮物件在影像中的位置與形狀，此特徵會和物件本身的色彩呈現高度相關，不同的影像可能畫出很相近的直方圖，而無法判斷影像中是什麼物件。而影像的光度變化，也會導致畫出的 Color Histogram 不同。

### B. HoG

此特徵全名為 Histogram of Oriented Gradients。在影像中，局部目標的形狀可以被邊緣的方向分布描述，具體的計算方式為：將影像切為多個可重疊的小區塊，稱為 Cell，接著再選取周圍的幾個 Cell，把每個 Cell 中各個像素的邊緣方向繪製成直方圖，這些直方圖結合後，就可以代表一個 Block。由於在計算邊緣時會受到影像的明亮度與陰影的影響，因此再對 Block 做對比度的 Normalization，如此可以降低明亮度太高的不良影響，同時考慮更多的像素也能避免獲得的特徵值都只考慮單一像素，能降低雜訊干擾。

此特徵要設定 Cell、Block 的大小，除了計算量龐大之外，最後得到的特徵也是一個維度很大的向

量，在後續訓練模型時會耗費極大量的時間。此外，若有遮蔽物擋住某些物件，計算出的邊緣方向也會受到影響。

### C. Haralick texture

此特徵利用灰階共生矩陣的原理，可以描述物件的紋理。物件的紋理在影像中會重複出現，將影像轉為灰階後，像素的灰階值會有反覆的變化。因此，相隔某段距離的兩個像素可能會有比例關係，將這種關係以矩陣的形式表達，即構成對應的灰階共生矩陣，可以用於尋找影像中重複出現的紋理。

Haralick texture 會計算灰階共生矩陣的一些統計量，做為影像的特徵，包含對比度 (矩陣內數值的分布和局部變化)、能量 (矩陣內每個數值的平方和)、..... 等等共 14 個特徵，由於第 14 個特徵在數學計算上有不確定性，因此一般選用前 13 個特徵，以描述影像。

此特徵不需要輸入額外的參數就能計算，且最後得到的特徵維度是 13 維，相較於前兩個特徵可說是非常的簡潔，在後續輸入模型做訓練所需要的時間也大幅降低；缺點是當資料量很龐大時，僅用 13 個特徵並不能很完善的描述所有的影像。

## III. Experiment

本次實驗的資料共有訓練集 63325 張影像、驗證集 450 張、測試集 450 張，這些影像包含 10 個類別。先將影像都縮放成  $128 \times 128$  後，再計算特徵。特徵計算方式為上一個段落所介紹的三個方法，相關的超參數設定呈現在下表，使用的模型是 AdaBoost、XGBoost、LightGBM，模型的部分並未特別挑選超參數。評估的方式使用 Top-1 accuracy、Top-5 accuracy、訓練時間。

Method	Setting	Dimension
Color Histogram	Bins of R、G、B = 8	512
HoG	Window size = $128 \times 128$ Block size = $64 \times 64$ Block stride = $32 \times 32$ Cell size = $32 \times 32$ Bins = 8 nlevels = 64	324
Haralick	-	13

Table 1: Setting of Feature Extractor

除了使用機器學習的方法，也嘗試使用深度學習的方法。在有關影像的任務中，CNN 是一個常用的做法，這裡嘗試搭建一個結構較簡單的 CNN，直接將影像輸入模型做學習。模型的設定呈現在下表。

Layer	Setting
Convolution1	in channels = 3
	out channels = 32
	kernel size = 5
Convolution2	in channels = 32
	out channels = 32
	kernel size = 5
Maxpooling1	kernel size = 2
Maxpooling2	kernel size = 2
Linear1	in features = 32*29*29
	out features = 64
Linear2	in features = 64
	out features = 10

Table 2: Setting of CNN

Model	Feature Extractor	Validation	Test
AdaBoost	Color Histogram	0.102 $\pm$ 0	0.108
	HoG	0.113 $\pm$ 0	0.08
	Haralick	0.078 $\pm$ 0	0.107
XGBoost	Color Histogram	0.091 $\pm$ 0	0.098
	HoG	0.108 $\pm$ 0	0.113
	Haralick	0.096 $\pm$ 0	0.096
LightGBM	Color Histogram	0.082 $\pm$ 0	0.078
	HoG	0.113 $\pm$ 0	0.104
	Haralick	0.093 $\pm$ 0	0.102

Table 3: Top-1 Accuracy

上表是使用機器學習模型搭配不同的特徵計算方式，以 Top-1 accuracy 評估模型對影像分類的結果。根據表格內容，Validation 的結果是使用 AdaBoost 搭配 HoG 與 LightGBM 搭配 HoG 有最高的準確率，而 Test 的結果，則是 XGBoost 搭配 HoG 有最高的準確率。

針對不同的特徵計算方式，最好的組合分別為 Color Histogram 配 AdaBoost、HoG 配 XGBoost、Haralick 配 AdaBoost。

Model	Feature Extractor	Validation	Test
AdaBoost	Color Histogram	0.478 $\pm$ 0	0.496
	HoG	0.52 $\pm$ 0	0.467
	Haralick	0.526 $\pm$ 0	0.52
XGBoost	Color Histogram	0.462 $\pm$ 0	0.533
	HoG	0.491 $\pm$ 0	0.491
	Haralick	0.551 $\pm$ 0	0.487
LightGBM	Color Histogram	0.5 $\pm$ 0	0.467
	HoG	0.491 $\pm$ 0	0.511
	Haralick	0.48 $\pm$ 0	0.498

Table 4: Top-5 Accuracy

上表是使用機器學習模型搭配不同的特徵計算方式，以 Top-5 accuracy 評估模型對影像分類的結果。根據表格內容，Validation 的結果以 XGBoost 搭配 Haralick，模型的分類效果最好，Test 的結果則是 XGBoost 搭配 Color Histogram 的分類準確率最高。

針對不同特徵，剛好都搭配不同的模型，在 Test 會有很好的效果，分別是 Color Histogram 配 XGBoost、HoG 配 LightGBM、Haralick 配 AdaBoost，沒有哪個

特徵可以搭配所有模型都得到很好的分類效果。

Model	Feature Extractor	Training Time
AdaBoost	Color Histogram	28.0 $\pm$ 0.035
	HoG	150.756 $\pm$ 1.764
	Haralick	7.082 $\pm$ 0.006
XGBoost	Color Histogram	115.49 $\pm$ 0.144
	HoG	546.152 $\pm$ 1.075
	Haralick	36.471 $\pm$ 0.047
LightGBM	Color Histogram	3.248 $\pm$ 0.107
	HoG	19.514 $\pm$ 0.042
	Haralick	1.098 $\pm$ 0.032

Table 5: Training Time

上表是使用機器學習模型搭配不同的特徵計算方式，紀錄每一種組合在訓練時需要花費的時間。根據表格內容，在模型的部分，時間由低到高依序為 LightGBM、AdaBoost、XGBoost；比較不同的特徵計算方式，由於 Haralick 的特徵維度為三者最少，因此放進三個模型時訓練都相當快速，其次是 Color Histogram，最慢的是 HoG。

Model	Validation Top1/Top5	Test Top1/Top5
CNN	0.1 $\pm$ 0 / 0.5 $\pm$ 0	0.100 / 0.5

Table 6: CNN

上表是使用 CNN 對影像做分類，並使用相同的指標評估效果。根據上表，我們可以看到在 Validation 與 Test 的結果都和前面使用的機器學習方法很接近。

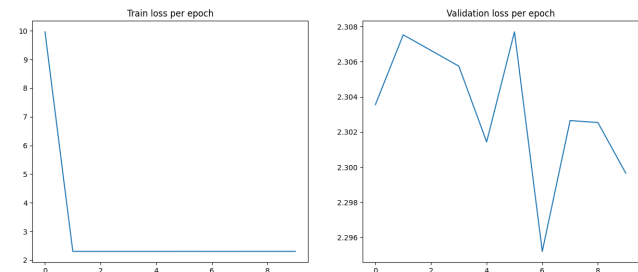


Figure 1: Loss for Training and Validation

上方是將不同 epoch 的 training loss 與 validation loss 視覺化的摺線圖。根據左圖，training loss 從第 2 個 epoch 後就開始維持定值；在右圖中，validation loss 則沒有隨著 epoch 增加而收斂到一個定值，在第 6 個 epoch 時最低。

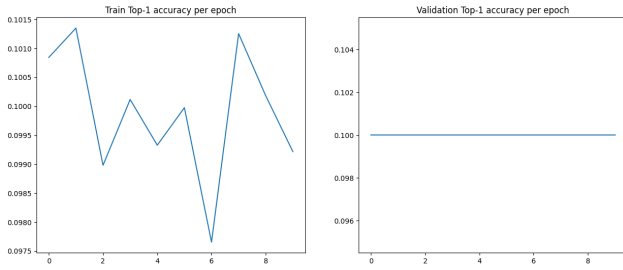


Figure 2: Top-1 accuracy curve for Training and Validation

上方是將不同 epoch 的 training top-1 accuracy 與 validation top-1 accuracy 視覺化的摺線圖。在左圖中，training 的 top-1 accuracy 會隨著 epoch 震盪，並沒有在固定的範圍收斂，在第 1 個與第 6 個 epoch 時分別有最高和最低的 accuracy，而右圖中的 validation top-1 accuracy 則一直都固定在 0.1。

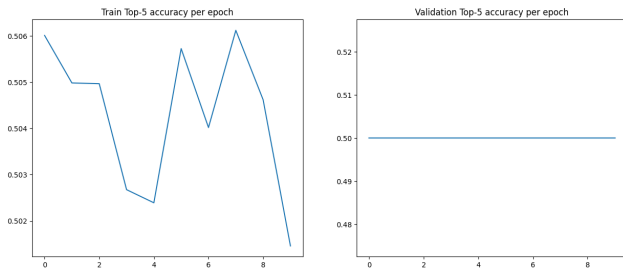


Figure 3: Top-5 accuracy curve for Training and Validation

上方是將不同 epoch 的 training top-5 accuracy 與 validation top-5 accuracy 視覺化的摺線圖。根據左圖，training 的 top-5 accuracy 也不會收斂到某個範圍，且震盪的情形和 top-1 accuracy 不同，分別在第 7 個與第 9 個 epoch 時有最高和最低的 accuracy；右圖中的 validation top-5 accuracy，則完全沒有變動。

## IV. Conclusion

本次作業中，我們使用多個方法分別對影像萃取出不同的特徵，並使用常見的機器學習模型學習特徵後，對驗證集與測試集的影像做分類。最後，也嘗試使用深度學習的 CNN 模型，並比較這些方法在 Top-1、Top-5 accuracy 的數值。根據實驗的結果，並沒有一種組合可以在兩種評估方式都超越所有的組合。除了比較分類準確率，我們也計算各個組合所需要的訓練時間。

在 CNN 的部分，由於這是第一次自己建立 CNN，對內部的參數功能還不太理解，訓練的結果也不盡理想。在確認 data loader 時，在同一個 mini batch 中的資料似乎都相同，導致在驗證階段的 accuracy 都一樣。除此之外，我不確定 training accuracy 的計算方式，是除以整個 training set 的數量，還是除以一個 mini batch 的數量。這些問題，都是我之後可以改進的方向。