
程 式 語 言

期 末 結 案 報 告

2019/06/14

資工 2A

張秉洋

106503014

壹、題目

此期末專題的題目是 *Object Detection*，利用 MatLab 中的 “Deep Learning Toolbox™” 來實作簡單的物品辨識與定位。同時藉此初步了解電腦視覺、機器學習、卷積神經網路(CNN)的原理與運用方式。

貳、工作進度安排

由於我是一個人一組，因此所有的工作皆是自行完成。至於進度安排的部分並沒有完全的照期初的提案報告，因當初低估了開始訓練前的準備工作所需的時間，不過最終結果仍有如期完成。

參、訓練細節

1. 使用的網路架構

本次使用的網路架構為 *Faster RCNN*，而所使用的 pre-trained model 為 VGG19 和 ResNet50，兩者分別用在 Single Label 與 Double Label。

原先是希望兩者運用相同的 pre-trained model，因 VGG19 在 Single Label 的表現不錯，但在 Double Label 的準確度非常的低，幾乎抓不到任何的東西。而在我嘗試使用 VGG16, AlexNet, GoogleNet, ResNet18 之後仍然無法改善。

而最後嘗試 ResNet50 之後發現 Double Label 的準確度可以達到八成以上。但相較於 VGG19，ResNet50 需要非常長的 detect 時間(在我的電腦上兩者至少差了快 2.5 倍)，且既然對 Single Label 來說準確度差不多的情況下我覺得都使用 ResNet50 的話效益太差，因此最後決定兩者使用不同的 pre-trained model。

2. 資料處理

我將資料 shuffle 後分為八成訓練，二成測試。如下圖的程式碼。

```
shuffledIdx = randperm(height(AllData));  
idx = floor(0.8 * height(AllData));  
trainingData = AllData(shuffledIdx(1:idx),:);  
testData = AllData(shuffledIdx(idx+1:end),:);
```

3. Networks' Options

因使用了不同的 model，因此以下也會分成 Single & Double 兩個部分。且兩個參數都是在不停的重複訓練與觀察訓練過程之後得到的較佳設定。

(i) Single Label

```
options = trainingOptions('sgdm',...  
    'InitialLearnRate',0.0001,...  
    'MiniBatchSize',5,...  
    'MaxEpochs',5,...  
    'CheckpointPath',tempdir,...  
    'ExecutionEnvironment','gpu');
```

(ii) Double Label

```
options = trainingOptions('sgdm',...  
    'InitialLearnRate',0.0001,...  
    'Verbose',true,...  
    'MiniBatchSize',5,...  
    'MaxEpochs',8,...  
    'Shuffle','every-epoch',...  
    'VerboseFrequency',30,...  
    'CheckpointPath',tempdir,...  
    'ExecutionEnvironment','gpu');
```

說明：在 options 的部分大多都是參照官方的範例文件，主要有作調整的部分為 InitialLearnRate, MiniBatchSize, MaxEpochs。原先在 MiniBatchSize 的部分範例文件預設為 10，使得剛開始訓練時不停的 out of memory，直到我詢問大三的學長之後才得知可以把 MiniBatchSize 改小一點來解決此問題。而 MaxEpochs 的部分對訓練的時間有很大影響，經過測試，Epochs 在大於目前設定的值時對於準確度沒有太大的提升，總訓練時間卻增加很多，所以選目前的值最為最終結果。最後 InitialLearnRate 的部分則是詢問助教後得知當訓練出的準確度不高時可以把 rate 調小一點對於結果會有幫助，所以從原先的 0.001 縮小了十倍變為 0.0001。

4. 準確度

在準確度的部分我用分割的測試資料連計算準確度，而 Single Label 為 89%，Double Label 為 85% (Double 若只有對一張圖算對一半)。

原先想要像是範例程式一樣顯示訓練的 training progress 來計算準確率，但是 Faster-RCNN 似乎不支援此方式(如下圖)，所以只好改成用測試資料計算。

```
options = trainingOptions('sgdm', ...  
    'MaxEpochs', 5, ...  
    'MiniBatchSize', 5, ...  
    'InitialLearnRate', 1e-4, ...  
    'CheckpointPath', tempdir,...  
    'Plots','training-progress',...  
    'ExecutionEnvironment','gpu');
```

※所設定之 options 參數

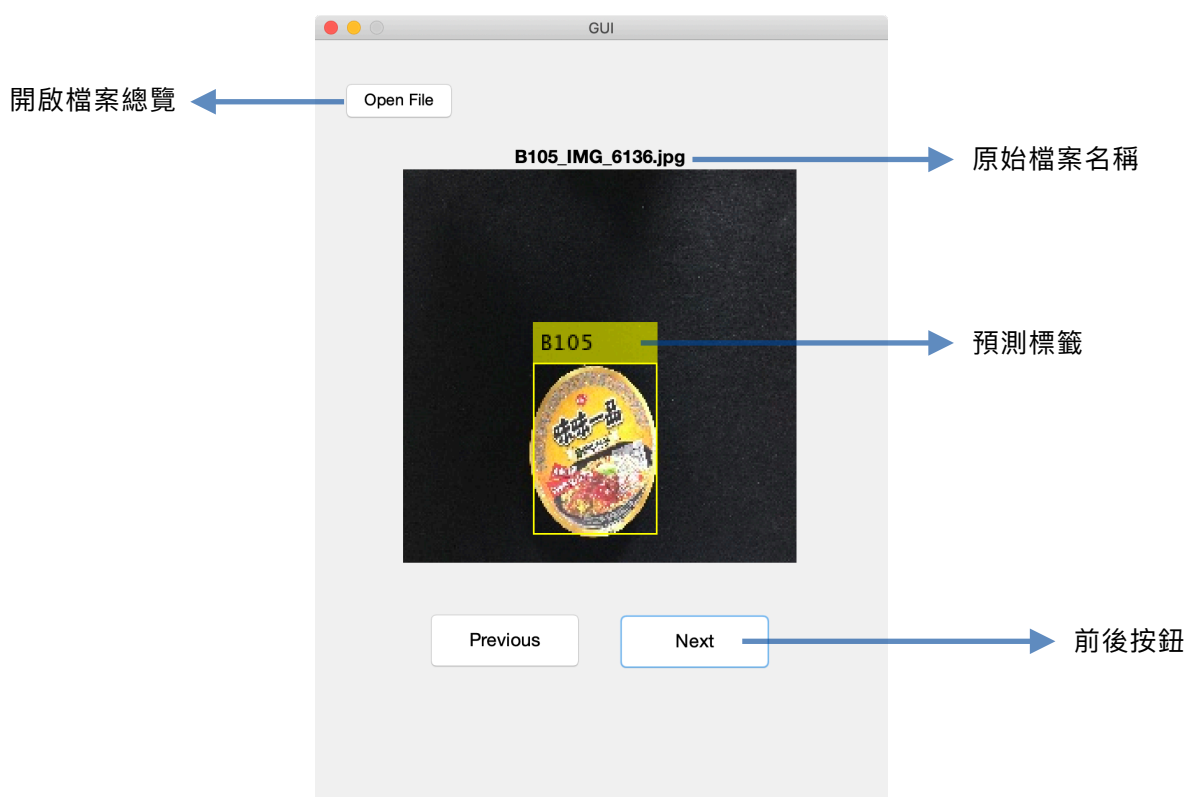
```
Error using vision.internal.cnn.validation.checkTrainingOptions (line 9)  
trainFasterRCNNObjectDetector does not support the training-progress Plots training option.
```

※出現的錯誤訊息

*確認並非打錯字的問題，因為若是打錯字的話在 run options 的部分就會跳出錯誤訊息。

肆、GUI 設計

此部分參照助教示範的方式，分別做有 Next 與 Previous 的按鈕讓使用者可以選擇。而介面的部分盡量保持簡潔明瞭，讓預測的結果與實際標籤一目瞭然。



伍、心得

在這次的專題中我覺得最花時間的不是訓練的過程，反而是事前的資料查詢與訓練資料的處理。在剛開始時光是要用哪種 CNN 的架構就花了多時間研究，既不會自己架設 CNN from scratch，也不是很清楚如何修改 Transfer Learning 的最後一層 Layer，直到最後選擇用 *Faster-RCNN* 才算是解決此問題。而在資料處理的部份，多虧了助教的協助提供範例程式碼，不然原先我自己做了超過一個禮拜仍然無法做出 RCNN 所接受的 input 格式。也就因為這兩個問題，讓我的進度大幅落後了期中報告的進度安排。

而在訓練的過程中也算是做了非常多的嘗試，用了各種不同的 Pre-trained model、參數，才有了目前八成的準確度。雖然不像其他組有九成五以上的準確，但相較剛開始訓練時 Double Label 不到一成準確度的慘況，我覺得這些嘗試非常的值得。

最後在經過這次的專題讓我瞭解到非常多關於機器學習方面的知識，也感受到要做深度學習的事前準備非常繁雜，此次多虧助教所提供的資料庫所以不用做標籤的定位，光是想到要手動定位 2000 多張圖片當作訓練資料我就覺得非常的可怕(雖然 2000 張圖片根本算是非常小的資料量)。總的來說，此次的練習讓我更想進一步學習如何使用深度學習來解決問題。