

Machine Learning Final Project Report

RSNA Pneumonia Detection Challenge (DeepQ Challenge)

隊名：NTU_r07943107_

隊員：r07943107 徐晨皓、b03901013 江庭瑋、b04505028 林秀銓、b04505034 呂冠蓉

● Introduction & Motivation (0.5%)

通過檢查胸部 X 光片 (Chest **X-Ray**, CXR) 來準確診斷肺炎 (Pneumonia) 是一項極為複雜的工作，因為它需要一群訓練有素的專家對於肺內各種狀況進行複雜的分析，進而診斷病患是否罹患肺炎。然而，由於深度卷積神經網絡 (DNN) 在近幾年的迅速發展以及對 CXR 進行的大量診斷研究，使得人工判斷病患是否罹患肺炎出現一線曙光。若在未來此技術發展成熟，不僅可以節省放射科醫師的勞力成本，也能降低人為判斷錯誤的機率。

本問題旨在使用深度學習之方法，在一張給定的胸部 X 光片中，判斷病患是否有肺炎之徵兆。若辨識為此病患患有肺炎，則將此 X 光片標為有疾病，且需將有肺炎徵兆的區域以方框圈出；反之，若辨識為此病患無肺炎病症，則將此 X 光片標為正常。

● Data Preprocessing/Feature Engineering (1.5%)

在資料前處理 (Data Preprocessing) 的部份，我們採用標準化 (Normalization) 及圖片縮放 (Image resizing) 兩技術。以下對此兩技術分別說明。

1. 標準化

為了優化梯度下降法 (Gradient descent method)，我們採用標準化來提高訓練精準度。首先，將原來 RGB 圖片轉成 BGR 圖片。接著，由於我們使用 keras 的 ImageNet 預訓練模型 (Pre-trained model) 來作為模型骨幹 (Backbone)，因此我們使用 keras 說明文件 [1] 中的標準化方法 *caffe*。其方法是將 BGR 圖片以下列方法轉化：

$$P'[\dots, 0] = P[\dots, 0] - 103.939,$$

$$P'[\dots, 1] = P[\dots, 1] - 116.779,$$

$$P'[\dots, 2] = P[\dots, 2] - 123.680,$$

where P and P' are the original and transformed images, respectively.

2. 圖片縮放

原始的圖片的尺寸為 1024x1024 之 RGB 圖片，如果直接用原始圖片去訓練模型，其所包含的資料量太大，不但會需要非常久的訓練時間，且分數並沒有明顯提昇，代表其中有一些不重要或是可忽略的資料，使得訓練過程很沒效率。因此，我們將原始的圖片尺寸縮小為 224x224 之 RGB 圖片，這不僅縮短訓練時間，而且分數與圖片縮放前差不多，有效提昇我們訓練模型的效率。

● Methods (2.5%)

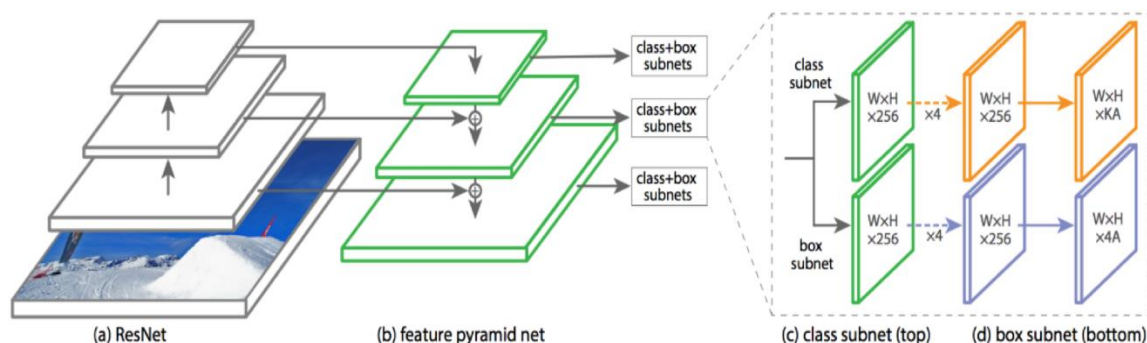
1. 神經網路架構：RetinaNet

對於物件辨識(object detection)問題，目前最先進(state-of-the-art)的技術有 YOLO (You Only Look Once) [2, 3, 4]、R-CNN (Regions with CNN features) [5, 6, 7] 與 RetinaNet [8]。我們選用第三種方法 RetinaNet，因其具有單階段(one-stage)方法架構簡單又速度快的優勢，且能夠達到同 R-CNN 等兩階段(two-stage)方法的高精確度。

RetinaNet 是由 Facebook AI Research (FAIR) 於 2017 年提出的一種網絡架構，運用關注損失(focal loss, FL)此損失函數解決了訓練過程中前景(foreground)和背景(background)比例極度不平衡，進而導致損失函數(loss function)來源大多被容易歸類的背景所佔據的問題[9]。One-stage 方法少了提議階段(proposal stage)，直接在特徵提取圖(extracted feature map)上每個位置輸出分類結果。由下列公式可以看到，FL 在原先的交叉熵(cross entropy, CE) 式子中新增一個關注參數(focusing parameter)，控制 $\gamma > 0$ 即可大大降低背景的損失(loss)貢獻，使得損失集中在較難分類的前景上。

$$CE(p_t) = -\log(p_t)$$
$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t)$$

RetinaNet 本身是由 ResNet、特徵金字塔網路(feature pyramid network, FPN)、分類子網路(class subnet)、方框子網路(box subnet) 四個子網路形成的。首先，ResNet 先使用 ImageNet 資料預訓練(pre-train)過，負責特徵提取(feature extraction)。FPN 作骨幹(backbone)，不同的金字塔層(pyramid level)對應不同大小的特徵圖(feature map)以處理各種大小物體的偵測。每一層的特徵圖都接上分類子網路與方框子網路，皆為全連結(fully-connected)卷積神經網路(convolutional neural networks)：前者使用 FL 作為損失函數，負責分類；後者負責找出定界框(bounding box)位置及大小 [9]。下圖為 RetinaNet 結構示意圖[8]。



至於 RetinaNet 的實現，我們使用由 Fizyr 所開發之 keras-retinanet 套件[10]。基於 keras 架構上，此套件包含 RetinaNet 模型和 FL 函數可以在訓練過程中使用，也包含其他物件辨識所需的相關前處理(preprocess)與回調函數(callback functions)。

2. 訓練方法

我們的方法主要參考自 Kaggle RSNA Pneumonia Detection Challenge 競賽中獲得第三名的 Phillip Cheng 之作法[11]。其訓練過程修改自 keras-retinanet 套件中的 train.py。

a. 訓練資料

訓練資料使用全部 21764 筆圖片資料，沒有將資料分割為訓練(training)和評估(validation)資料。亦沒有用到 train_metadata.csv 檔案。訓練標記(training labels)來自 train_labels.csv，辨識的分類(class)分成兩種：有肺炎(1)和無肺炎(0)，有肺炎的將會註解座標資訊(annotation)。

在資料增強(data augmentation)方面，使用 keras-retinanet 套件中之 CSVGenerator 生成圖片以增加訓練資料量。Generator 轉換設定如下：

- i. 位移(translation)：從(-0.1, -0.1)至(0.1, 0.1)。
- ii. 縮放(scaling)：從(0.8, 0.8)至(1.2, 1.2)。
- iii. 翻轉(flip)橫軸(x 軸)：機率為 0.5。

b. 訓練模型

模型使用 RetinaNet，分別嘗試 Resnet-50、Resnet-101、Resnet-152 作為骨幹，並如前章節所提，模型權重(weights)已經 ImageNet 圖片預訓練過。在損失函數方面，FPN (Regression submodel)採用 smooth L1，而分類子網路(classification subnet)的部份則採用 FL。在優化器方面(optimizer)，參考 FL 該

篇論文使用 SGD 方法，參數設定學習率(learning rate)為 0.01、動量(momentum)為 0.9、衰退率(decay)為 0.0001、nesterov=True[8]。

訓練過程將批次大小(batch size)設為 8，每一個 epoch 有 2500 步數(steps)，訓練 25 個 epoch。在回調函數中加入 ReduceLROnPlateau，當損失(loss)停止下降 4 個 epoch 時，即降低學習率 0.9 倍。硬體使用 NVIDIA TITAN V，每訓練一個 epoch 便將模型儲存下來。

3. 預測方法

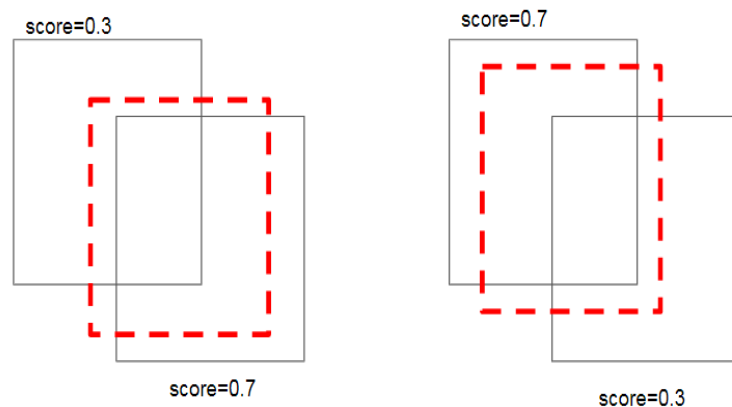
a. 模型預測

輸入訓練好的模型，將之轉換為推論模型(inference model)，輸入經預處理後的 4998 筆測試資料(testing data)，輸出有無肺炎標記(labels)、肺炎定界框(bounding box)位置與定界框之信心分數(confidence scores)。信心分數的閾值(threshold)設定為 0.04。

b. 改進方法

我們參考此篇[11]之作者發現預測出來的定界框過大，將之縮小(shrinking)將會有更好的預測準確率。因此我們嘗試不同的縮小倍率，並觀察其表現結果。

另外，我們亦嘗試將不同種類 ResNet 骨幹所訓練得到的模型整合(Ensemble)起來。當擁有多個模型預測，必須處理重疊的 bounding box 問題。原作者使用了 Non-maximum suppression 和 weighted averages(如下圖)的技巧，並且藉由調整 solo_min 參數選取信心分數高於此閾值之定界框。

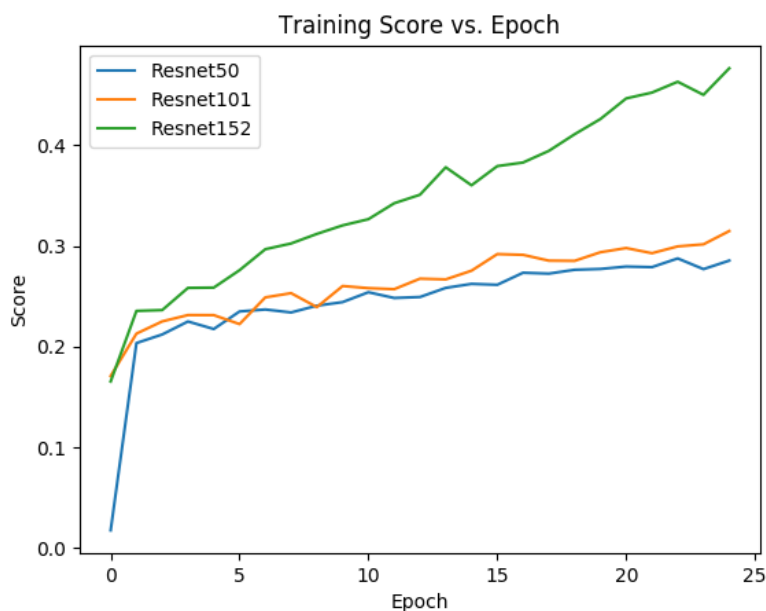


● Experiments and Discussion (4%)

1. Private & public scores

Backbone	Public score	Private score	# parameters	Model size (MB)
Resnet50	0.28161	0.21075	36,403,702	278
Resnet101	0.26735	0.22468	55,448,054	424
Resnet152	0.25843	0.20506	71,137,782	544

2. Training score vs. epoch



由上圖可看出，使用 Resnet-50 作為 backbone 之 RetinaNet 的 training score 較使用 Resnet-152 及 Resnet-101 的 training score 低。這可能是由於參數量多的模型比參數量少的模型，更容易 fit 訓練資料。然而，參數量多的模型容易 overfit 訓練資料，因此 training score 高並不代表 testing score 也會高。

3. Shrinking factor vs. accuracy

Shrinking factor	Public score	Private score
0.00	0.21275	0.20126
0.10	0.25994	0.22025
0.15	0.27201	0.22405
0.17*	0.27544	0.22215
0.20	0.27407	0.21772
0.25	0.26529	0.20632
0.30	0.23978	0.18607
0.35	0.20713	0.16582
0.40	0.17009	0.14367
0.45	0.14238	0.13417
0.50	0.12524	0.12658
0.60	0.11673	0.12215
0.70	0.11659	0.12025
0.80	0.11659	0.12025
0.90	0.11659	0.12025
1.00	0.11659	0.12025

(*使用於最佳模型中)

由上面的實驗數據可以觀察到，若直接將預測出來的定界框座標當作最終的預測結果(縮小率為 0)，無論 public score 或是 private score 皆沒有達到最好的結果，而是將定界框稍微縮小 0.15 倍，結果反而更加精確，但是若是縮小超過 0.3 倍，則有些異常的區塊會因為定界框的範圍太小而沒有被預測出來，造成 scores 大幅下降。

4. Solo_min vs. accuracy

solo_min	Public score	Private score
0.10	0.26364	0.20253
0.15*	0.28161	0.20253
0.20	0.27503	0.21202

0.30	0.25322	0.18291
0.40	0.22427	0.15253
0.50	0.19108	0.14303

(*使用於最佳模型中)

若預測出來的某一個定界框沒有與其他定界框交集，則我們會利用 solo_min 把信心分數較低的定界框排除掉，而經過實驗，solo_min 為 0.15 至 0.2 時分數最好，若再將 solo_min 提高，則會將一些正確的定界框排除掉，造成分數有明顯下降。

5. Ensemble

wt_overlap	Public score	Private score
0.00	0.23895	0.18987
0.10	0.23895	0.18987
0.20	0.23895	0.18987
0.30	0.23895	0.18987
0.40	0.23936	0.1924
0.50	0.24279	0.1924
0.60	0.24115	0.19177
0.70	0.24019	0.19177
0.80	0.24567	0.19493
0.90	0.25198	0.21265
0.95	0.26227	0.21518
1.00	0.2786	0.22468

此實驗是用兩個模型來實作 ensemble，一個為以 Resnet-50 為骨幹，另一個以 Resnet-101 為骨幹。若兩個模型所預測出來的定界框交集的面積與定界框的面積的比率大於參數 wt_overlap，則會被將所有有交集的定界框做加權平均，得到一個新的定界框作為最終的預測結果，因此 wt_overlap 越高，代表定界框需要交集面積越大，交集的區塊越有可能是有異常的，而經由實驗可以觀察到分數確實越來越高。

● Conclusion (1%)

本報告使用 RetinaNet 來判斷給定的胸部 X 光片有無肺炎之病徵，若判斷為有罹患肺炎，則使用若干矩形將肺炎區域框出。我們嘗試使用不同的預訓練模型作為模型骨幹，包括 Resnet-50、Resnet-101 以及 Resnet-152，並探討其訓練過程的不同與預測準確率的差異。另外，我們也設計實驗觀察不同的縮小參數(shrinking factor)與不同的閾值(Solo_min)對於預測準確率有何影響。最後，我們探討集成(ensemble)模型的效果如何。

● References (0.5%)

- [1] Reference implementations of popular deep learning models. <https://github.com/keras-team/keras-applications>
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," In *CVPR*, 2015.
- [3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," In *CVPR*, 2017.
- [4] J. Redmon and A. Farhadi. (2018). "YOLOv3: An incremental improvement." [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," In *ECCV*, pages 346–361, 2014.
- [6] R. Girshick, "Fast R-CNN," In *ICCV*, 2015.
- [7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," In *NIPS*, 2015.
- [8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," In *ECCV*, 2017.
- [9] 來自 FACEBOOK RESEARCH 物體偵測的最新進展：FOCAL LOSS FOR DENSE OBJECT DETECTION. <https://data-sci.info/2017/08/10/%E4%BE%86%E8%87%AAfacebook-research%E7%89%A9%E9%AB%94%E5%81%B5%E6%B8%AC%E7%9A%84%E6%9C%80%E6%96%B0%E9%80%B2%E5%B1%95%EF%BC%9Afocal-loss-dense-object-detection-2/>
- [10] Keras-retinanet. <https://github.com/fizyr/keras-retinanet>
- [11] 3rd place solution in RSNA Pneumonia Detection Challenge. <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/discussion/70632#latest-440310>