

# Machine Learning Final Project Proposal

- 隊名及組員

1. 隊名

NTU\_r07943107\_

2. 隊員

r07943107 徐晨皓

b03901013 江庭瑋

b04505028 林秀銓

b04505034 呂冠蓉

- 題目

RSNA Pneumonia Detection Challenge (DeepQ Challenge)

- Problem Study

本問題旨在使用深度學習之方法，在一張給定的 X 光片中，偵測病患是否有肺炎 (Pneumonia) 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。

此問題為物件辨識(object detection)的任務。目前最先進(state-of-the-art)的物件辨識技術有的 YOLO [1, 2, 3]、RCNN [4, 5, 6]與 RetinaNet [7]。以下簡介此三種技術。

YOLO (You Only Look Once) [1]的作法是將一張圖片分割成很多方格(grid cell)，並對每一個方格進行預測，給出此方格對於每種 label 的信心指數(機率)，在 network 最後一層，根據這些機率進行 bounding box 的圈選以及 label。YOLO 使用的 Activation function 除了最後一層是線性輸出外，其餘各層皆使用 leaky relu 作為 Activation function。Loss function 方面，作者對於 grid cell 的分類錯誤以及 bounding box 的圈選錯誤都有特別的考量，如此一來的表現會比一般使用平方誤差和(squared-error sum)還要好。在 2015 年，YOLO 發表後，其團隊又再做進一步的改良，因此許多版本相繼被發表出來，如 YOLO9000 [2]、YOLOv2 [2]以及 YOLOv3 [3]，在更多樣的影像類別中的辨識成功率獲得大幅提升。神經網路的效率也提高，不僅預測的時間減少，訓練時間也減少，準確率更高。

RCNN (Regions with CNN features)的概念是利用 CNN 的特徵提取(feature extraction)與候選區域選擇(region proposal)來對影像進行分析，最後以邊界回歸(bounding-box regression)獲得圈選區域。SSP-Net [4]對於傳統 RCNN 的大量卷積(convolution)計算進行改良，提出空間金字塔池化 (Spatial Pyramid Pooling, SPP)，將 SPP 層置於卷積層後減少大量卷積計算時間，此方法有效的加速傳統 RCNN 的測試速度約 24~102 倍。爾

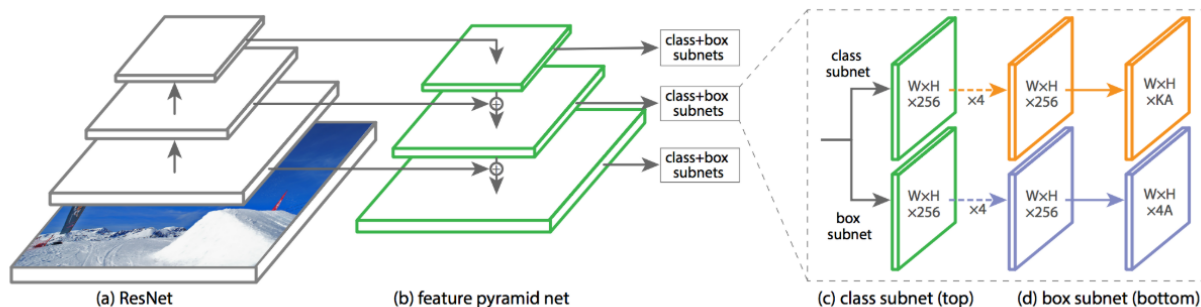
後，Fast-RCNN [5]與 Faster-RCNN [6]又相繼被提出，以解決 SPP-Net 的速度問題。Fast-RCNN 提出簡化版 ROI (region of interest)池化層，而不使用 SPP，使得網路能夠反向傳播(back propagation)，進而使整體網路一起訓練。另外，Fast-RCNN 以 softmax 取代 SVM，又以 SmoothL1Loss 取代邊界回歸，最後以 SVD 加速全連通層(fully connected layer)。Faster-RCNN 進一步將候選框選取合併到神經網路中，使得使用者無需預先設定候選框，而是讓神經網路學習如何選取，這技術又進一步加速 Fast-RCNN。

RetinaNet 則是由 Facebook AI Research (FAIR) 於 2017 年提出的一種網路架構[7]，有 one-stage method 速度快的優勢，並用 Focal Loss (FL)這種新的損失函數解決了前景和背景比例極度不平衡而導致 loss function 來源大多被容易歸類的背景所佔據的問題，擁有高精確度 [9]。One-stage 方法少了如上述 RCNN 等 two-stage 方法中的 proposal stage，直接在 extracted feature map 上每個位置輸出分類結果。由下列公式可以看到，Focal Loss 在原先的 Cross Entropy (CE) 式子中新增一個 focusing parameter，控制  $\gamma > 0$  即可大大降低背景的 loss 貢獻，使得 loss 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、feature pyramid network (FPN)、class subnet、box subnet 四個子網路形成的。ResNet 先用 ImageNet 資料 pre-train 過，負責 feature extraction。FPN 作為 backbone，不同的 pyramid level 對應不同大小的 feature map 以處理各種大小物體的偵測。每一個 level 的 feature map 都接上 class 和 box subnet，皆為 CNN fully-connected network：前者使用 focal loss 作為損失函數，負責分類；後者負責找出 bounding box 位置及大小。下圖為 RetinaNet 結構示意圖 [9]。



## ● Proposed Method

### 1. Data Preprocess

原始的圖片的尺寸為 1024x1024 之灰階圖片，然而我們認為這樣大尺寸的圖片中包含許多不重要且可忽略的訊息。因此我們先對原始的圖片做前處理，將圖片尺寸縮小為 256x256，這不僅使訓練過程更有效率，也沒有遺失太多重要資訊。另外，我們也會對圖片進行正規化(normalization)。

## 2. Network structure

我們所使用的 structure 為上文所提及的 RetinaNet，並預先載入 keras-retinanet 套件中 pretrain 好的 weights。在 loss function 方面，FPN 採用了 smooth L1 的方式，而分類的部份則採用 local Loss。至於 optimizer 方面我們先採用 SGD 的方法，預計之後會再試試其他的方式，如 adam 等等。另外，因為依照 hw3 的經驗，使用 image generator 可以有效提升預測的準確率，因此我們也會使用 image generator 來增加訓練的資料量。

- References

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," In *CVPR*, 2015.
- [2] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," In *CVPR*, 2017.
- [3] J. Redmon and A. Farhadi. (2018). "YOLOv3: An incremental improvement." [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [4] 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.
- [5] R. Girshick, "Fast R-CNN," In *ICCV*, 2015.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," In *NIPS*, 2015.
- [7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," In *ECCV*, 2017.
- [8] RCNN 介紹. <https://blog.csdn.net/xyfengbo/article/details/70227173>
- [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/>