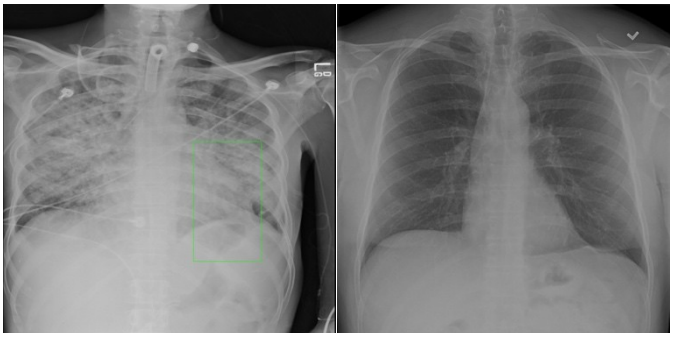
ML Final - Pneumonia Detection

隊長：b05901170 陳柏志 隊員：b06901004 劉穎立 b06901096 黃昱翰

Introduction & Motivation : 0.5%

本題是要 train 出一個可以分辨胸腔 X 光片(如圖一)是否患有肺癌的機器，也就是有分辨肺部沒有腫瘤，如果判斷出有患病，便再進一步將腫瘤部位用長方形框選出來。

我們會選這題，是因為作業沒有做過圖片中的物件框選，而這又是一項日常生活常見的技術，我們對它背後的原理感到好奇，因而選擇了這題。



（圖一）胸腔 X 光片及腫瘤部位匡選示意圖

Data Preprocessing/Feature Engineering : 1.5%

1. Picture pre-scaling

   我們將原本大小為 1024x1024 的圖片先做縮放，縮成 512x512，原因是如果把整張圖片用原大小丟進去 train，會 train 不太起來；而縮太小又會讓 performance 下降(此方法參考 RSNA Pneumonia Detection Challenge Kaggle 比賽的第二名的作法)，所以我們最後事先縮放成 512x512 在丟進 Retinanet 去 train。

2. Data augmentation

   我們也有將圖片進行基本的 data augmentation，但是我們部分的側姿勢有帶框框的(腫瘤的位置的長方形)，縮放可以讓框框跟著等比例縮放，但旋轉的話就無法讓框框跟著轉(他就不會是一個可以用長寬個其中一個點的座標可以表示的長方形)，而如果只有圖片轉框框沒有跟這轉的話，容易會讓原本的框框框到錯誤的地方，所以我們最後最多只轉到 6 度(此方法參考 RSNA Pneumonia Detection Challenge Kaggle 比賽的第二名的作法)。

Methods (At least two different methods) : 2.5%

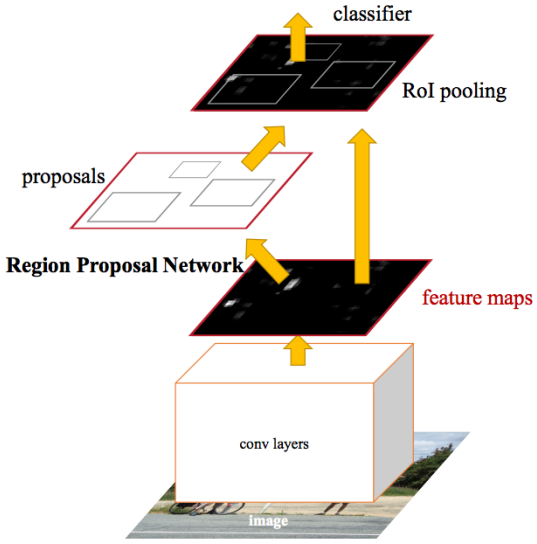
我們參考助教 PPT 中所介紹的三種方法，分別是 RCNN、YOLO 和 Retinanet，我們三種方法都有嘗試，而最後是選擇表現最好的 Retinanet，以下簡介我們所嘗試的三種方法：

1. RCNN

RCNN 是常見的一個圖像切割辨識的作法。傳統的 RCNN 是先用一些 image segmentation 法將圖片切塊，再過一些 pretrain 好的 model，便是該區塊是什麼物體。後來為了增加算速度等目的，又有 fast RCNN、faster RCNN 等類似概念的模型出現，本組剛開始是嘗試用 faster RCNN 的概念(如圖二)，加入一些我們的想法：先用 Selective Search 找出所有可能的框框，在用 CNN 決定他是腫瘤的機率。

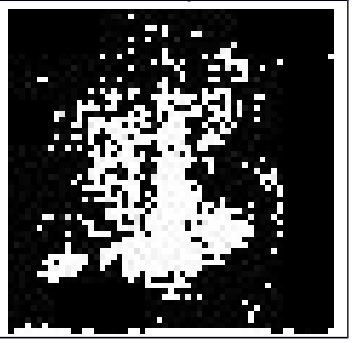
我們將 training data 分成兩部分，將原本的 1024x1024 的大圖，依照 csv 檔中的框框分成框框內跟框框外(腫瘤部分跟非腫瘤部分)，分別隨機切出好幾塊 128x128 的小圖，並標上 label(框框內腫瘤是 1、框框外非腫瘤是 0)，拿這些小圖跟 label 去 train 一個 CNN 的 binary classification model。再將 test 的原圖用類似 convolution 的方法，掃出很多 128*128 的小圖，拿去過前述 train 好的 classification CNN，並將結果做成一個由 1, 0 組成的 map(如圖三，白色部分代表 1，黑色部分代表 0)。

1 欲密集的地方便代表可能為腫瘤的機率愈高。然後用 map 去做 Selective Search 切出可能有東西的框框，再去算每個框框中單位面積的 1 的數量，作為他的分數，最後選擇左右半平面分數最高，且大於我們從 training data 中觀察而得的閾值的框框作為結果；如果兩邊皆沒有框框的分數大於閾值，則判定為沒有罹患肺癌。



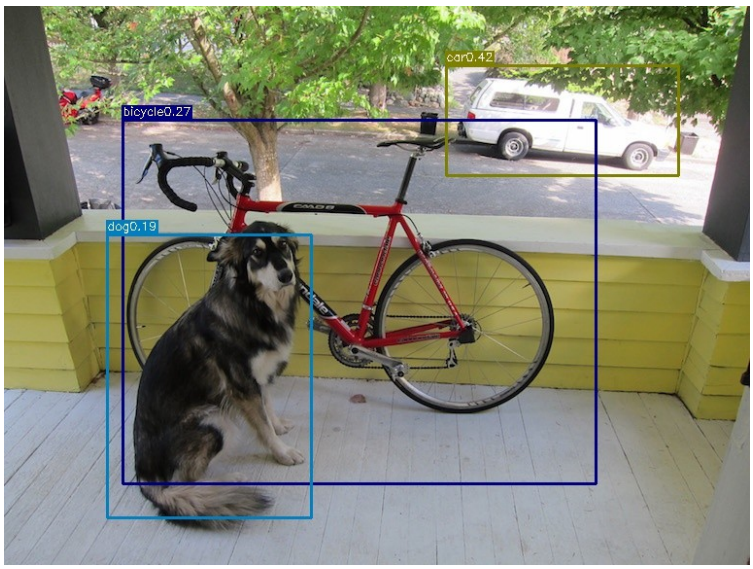
（圖二）Faster-RCNN 的概念圖[1]

1 圖片來源：https://medium.com/@syshen/%E7%89%A9%E9%AB%94%E5%81%B5%E6%B8%AC-object-detection-740096ec4540

原圖	Map
	

（圖三）用 RCNN 原理實作出的 feature map

2. YOLO



YOLO 是一個架構相對比較小的模型，而在訓練資料較少且沒有使用 pretrained model 的情況下，YOLO 仍可以有一定的表現。

其中我們將圖片調整為 512*512 的大小，將每張圖片進行 normalize，最後單純由 YOLO 得到最好的成績為 public: 0.21042 private: 0.17468. 但使用 YOLO 有諸多限制，例如沒有腫瘤的圖片無法加入訓練，及 model 尺寸太小，表現成績有限等。

原本的 YOLO setting:

- Model structure: In compared to the paper, I changed structure of top layers, to make it converge better. You could see the detail of my YoloNet in **src/yolo_net.py**.
- Data augmentation: I performed dataset augmentation, to

make sure that you could re-trained my model with small dataset (~500 images). Techniques applied here includes HSV adjustment, crop, resize and flip with random probabilities

- Loss: The losses for object and non-objects are combined into a single loss in my implementation
- Optimizer: Use SGD optimizer and my learning rate schedule is as follows:

Epoches	Learning rate
0-4	1e-5
5-79	1e-4
80-109	1e-5
110-end	1e-6

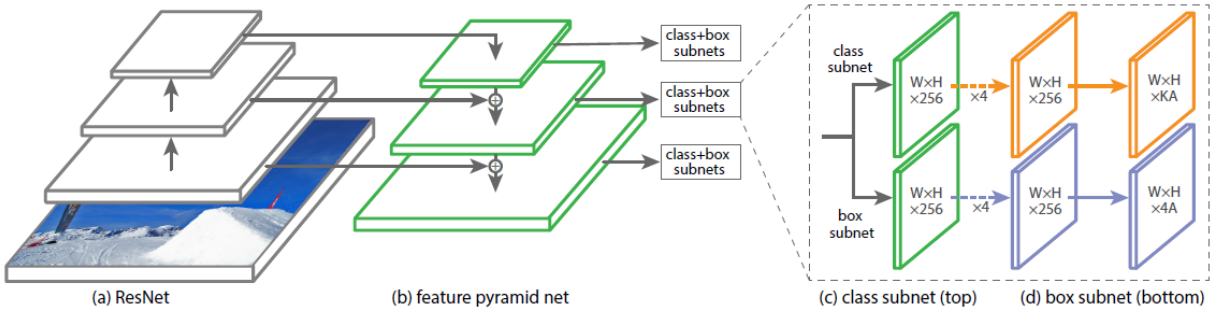
我們的改進之處：我們使用 adam optimizer 取代 SGD，再我們這個 case 中有稍微較好的表現。另外我們也實作的 data augmentation，將原始圖片隨機進行翻轉及小幅度旋轉。另外我們也在同一個腫瘤框框中再額外多框出幾個較好的框框，進行 data label 的 augmentation。

3. Retinanet

Retinanet 是一個 base on FPN 架構的物體檢測網路，其架構如圖四。他是將一個 pre-train 好的 Resnet CNN model 的 P3~P7 的 feature 抽出來，在丟進一些 subnet 作處理，他的重點是引進了 Focal Loss 這種新的計算 Loss 方式，不但可以調整正負樣本的權重，還可以提高難分辨樣本的權重。其表示方法如下：

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

我們便是基於 Restinanet 的方法，加入一些改變，並把 pre-scaling 後的圖片丟進去 train，再將結果接一個 CNN 的分類 model 得到結果，詳細的更改過程及結果在下一點會進行討論。



（圖四）Retinanet 基本結構概念圖[2]

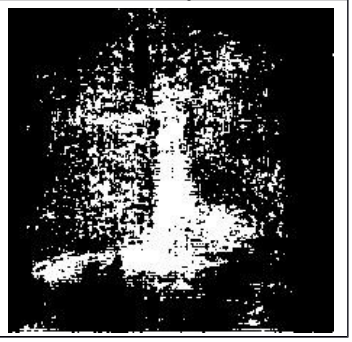
2 圖片來源：https://zhuanlan.zhihu.com/p/48958966

Experiment and Discussion：4%

1. RCNN

我們最先是用上免所敘述的，先 pre-train 好一個可以分辨 128x128 小圖是否為腫瘤的一部份的 CNN(CNN 架構參考 DCGAN)，再用他去掃原圖得到由 0, 1 組成的 feature map，在用 Selective Search 還找出可能的框框 。

我們有嘗試各種大圖片、CNN 的參數跟分數閾值，但後來因為分數一直上不去，連 simple 都過不了，所以我們便嘗試其他方法。我們認為應該是由於我們用還 train cnn 的測資是從原本的框框中隨機切出來的小圖，但是腫瘤並不是長方形的，框框中應有很大的部分是普通的肺，但可以確定的是它應該都會在肺裡。而根據結果觀察也發現，我們 train 出的 CNN 比起便是腫瘤，更像是一個可以辨識肺的機器(如圖五)，因次我們拿去框框框他也只會框選出它認為是肺的地方

原圖	Map
	

（圖五）解析度 512x512 的 feature map 與原圖比較

2. Retinanet

我們最先是嘗試直接把圖片原封不動丟進網路上找到的 Retinanet 架構去 train，發現不但一個 epoch 要 train 十幾個小時，而且也 train 不太起來(我們用 google colab 跑，他過一段時間就會重連)，因此我們嘗試把原本的模型做簡化，但是一直沒有成功。最後我們跑去爬 RSNA Pneumonia Detection Challenge Kaggle 比賽的前幾名的作法，發現第二名的作法跟我們正在做的事情最相近，因此我們基於他的作法再做修改：

(1)新增加 Classification layer

我們在 Retinanet 的後面接了一個 binary classification 的 Layer，根據 Retinanet 的結果作分類，區分罹患肺癌與否，以及最可能的腫瘤位置。

(2)修改 pre-train cnn model 為 Se-resnext101

RSNA Pneumonia Detection Challenge Kaggle 比賽第二名的報告中有說他嘗試了各種 resnet model 後，發現 Se-resnext101 的效

果最好，其次是 Se-resnext50，我們嘗試後也發現 Se-resnext101 的效果真的有好很多，因此我們最後是採用用 imagenet pretrain model 來作為我們 Retinanet 的 backbone。

(3) 多抽一層淺層的 Feature，讓他可以偵測到小物件

原論文中的 Retinanet 是用 FPN 架構抽取 P3~P7 的 feature(如圖四)，由於題中有一些腫瘤大小很小，而我們發現取愈淺的 layer 愈能辨識出小的物體，因此我們多往前抽了一層，去 P2~P7 共 6 層的 feature 來用，效果也有變好(詳情請見 model.py 中的第 46-47 行)，

(4) 增加 Dropout 的 layer

為了讓 classification 和 box regression 同時得到最佳的 model，我們加入了 dropout Layer，並發現表現有明顯進步

(5) 調整參數

最後一步不意外的是調參數，首先是 Focal Loss 中的 α 和 γ ，根據論文顯示 $\alpha=0.25, \gamma=2$ ，效果最好，我們測試結果也差不多，最後的結果是用上下微調後的結果去做 ensemble。

(6) Ensemble

我們所用的 ensemble 方法很單純，我們取了分數前幾高的 model 的預測結果。首先是是否患病，也就是結果中的 label，我們是只要有一半以上的 model 說他有病，我們就判斷他有病，並取說他有病的 model 中，單獨上傳 Kaggle 分數最高的一個的 BOX 作為最終的 BOX 結果。

Conclusion : 1%

我們最後使用 Retinanet 的方法，並基於 RSNA Pneumonia Detection Challenge Kaggle 比賽的第二名的作法作修改，主要試出有成效的方法有：

1. Picture Pre-scaling
   先將原圖做縮放後再餵進 model
2. Data Augmentation
   預先將圖片做旋轉縮放，增加樣本變異性
3. Se-resnext101
   更換 pre-train 的 resnet model，達到最好的表現
4. Add Classification Layer
   在 Retinanet 後面接上分類的 CNN Layer
5. Smaller Anchors Detect
   為了辨識出小範圍的物件(腫瘤)，我們用 FPN 架構多往上抽了一層 feature(P2~P7)
6. Add Dropout Layer
   為了讓 classification 和 box regression 同時得到最佳的 model

7. Ensemble
   最後挑選最好的數個 model 做 ensemble，分數可以再上升

Reference：0.5%

1. RSNA Pneumonia Detection Challenge Kaggle 比賽的第二名
   https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/discussion/70427
2. Retinanet
   https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4
3. RCNN
   https://medium.com/@syshen/%E7%89%A9%E9%AB%94%E5%81%B5%E6%B8%AC-object-detection-740096ec4540
4. YOLO：
   https://github.com/vietnguyen91/Yolo-v2-pytorch