

ADLxMLDS Final Project :

Weakly-Supervised Learning for Disease Detection

Team : 資管挖礦小鬼

B03705012 張晉華 B03705013 徐逸然 B03705014 陳威宇

1. Problem Definition

本次期末專題我們選擇做 HTC 的競賽題目。目標為建立一個機器學習模型，將一張 X 光照片讀入後可以自動分辨 X 光檢測結果可能含有那些病因，並進一步找出 X 光照片中病因發生之潛在部位。然而由於提供的 label 並無包含疾病之 bounding box，因此建立好 Classifier 後該如何找出 bounding box 是本次專題最大之挑戰。

2. Preprocessing/Feature Engineering:

資料的輸入為:

胸腔 X 光照片(1024*1024) 以及每張 X 光照片對應之疾病。

機器學習模型的輸出為:

胸腔 X 光照片可能包含那些病因，根據病因畫出可能的 bounding box。

資料 Preprocessing:

根據過往訓練圖像分類器的經驗，由於每張照片拍攝的對比度、亮度等不同，因此我們通常會考慮統一將訓練圖做 histogram equilization，這種方法可以用來增加圖像的局部對比度，尤其是當圖像的有用數據的對比度相當接近的時候。在肉眼觀察幾張 X 光照後，可以發現疾病出現的區域通常只會比周圍區域稍微白灰一些，有些甚至很難觀察的到，因此我們認為做 histogram equalization 可以更好的使疾病特徵顯現出來。

除此之外，由於原圖(1024,1024)的大小遠超出硬體設備可以負擔的訓練量，因此我們將照片 resize 成(256,256)，再不損失太多訓練精準度的情況下，大幅加快訓練速度。

3. Model Description:

我們的分類器採用 ResNet50 的架構，直接使用 keras.application 中 resnet50 做為 base layer，而經過我們幾次的實驗後，發現遵照 ResNet 原始的論文中在 flatten() 之後不增加任何的 Fully Connected Layer 就可以達到跟以往 VGG16 或 VGG19 接上三層全鏈層一樣的水準，甚至好上許多，模型的參數也因此下降，減少訓練時間。

分類器建立好之後，給定一個 classifier、一個 class 跟一個 image input，去找到在最大化 class 機率的目標下，image 每個位置的改變帶來的影響大小，並得出 model 的 saliency map，一個區域內值愈高代表對 class 的影響力愈大。在找 bounding box 時會將 saliency map 轉化成 0~1 之間的矩陣，大小則與 model input 一樣為 256*256，class 則由 classifier 的輸出提供，由於一個 case 可能有多個疾病，因此我們會把所有機率>0.5 的 class 都取出來計算 saliency map，在實際 implement 時使用了 keras-vis 這個套件，來做 saliency map 的計算。

計算 Bounding Box 的方法我們定義了以下幾種:

- Naive method

給定一個 threshold，找到所有大於 threshold 的點，取出 bbox 極端點
(Xmin,Ymin) ,(Xmax,Ymax)

- Split bbox

先用 Naive method 找到一個給定 threshold 的 bbox 當作初始值，再利用科西不等式不斷縮小，去找到一個可以平分 saliency map 總和的 bbox，使其包含的區域內 saliency map 分數總和和外框相等，藉此找到影響力較高的區域。

- Smooth bbox

給定一個 threshold 和一個 k 值，利用以下作法：

1. 將原本的 saliency map 各點值，改為上下左右和自己的平均值
2. 重複 k 次步驟 1.
3. 到達 k 次後，用 Naive method 找到 bbox

這個方法可以有效的踢出 saliency map 上的 outlier，正確的高影響力區域由於周圍的影響力也會高，因此在做完 k 次 iteration 後，其 saliency map 上的值不會下降太多，然而若是 outlier 被誤判成影響力很高的點，由於其周圍的點影響力不高，做完 k 次 iteration 後其 saliency map 上的值會大幅下降，因而不會納入 bounding box 候選點。

比較上三種作法上計算時間和複雜度的排序由高而低為：

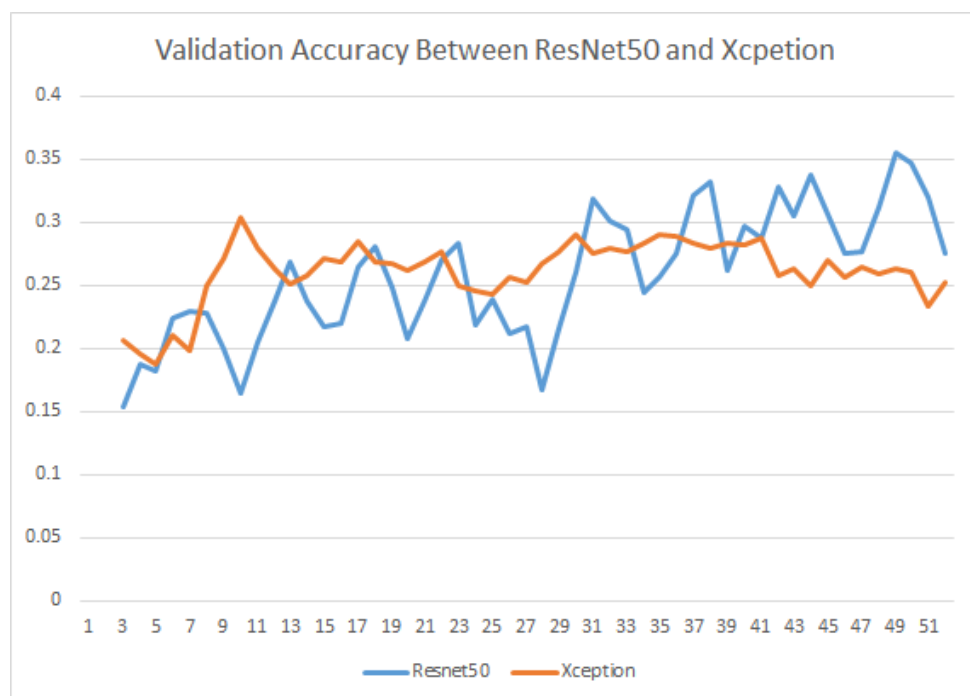
Smooth bbox > Split bbox > Naive method

三種作法所找到的區域平均影響分數排序也與上相同，由於模型的圖片輸入為 256×256 ，因此最後的 bounding box 位置與長寬需要再 $\times 4$ ，才是在 1024×1024 圖片上的座標和長寬。

4. Experiments and Discussion:

分類器模型調整:

我們嘗試用了 ResNet50 與 Xception 兩個架構來訓練，以下附上訓練數據作圖。由於 Validation set 很小，Accuracy 的數值變動很大，因此我取了 3-moving average 以方便比較。可以看見 Xception 架構的準確度隨著訓練 epoch 增加並沒有太大的變動，ResNet50 則能明顯看出上升趨勢，再加上訓 Xception 訓練時間高出 ResNet50 30%左右，因此最後選擇使用 ResNet50 做為分類器架構。



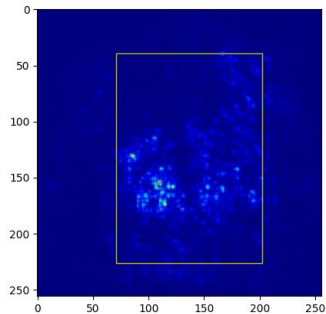
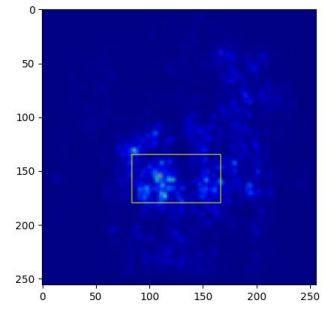
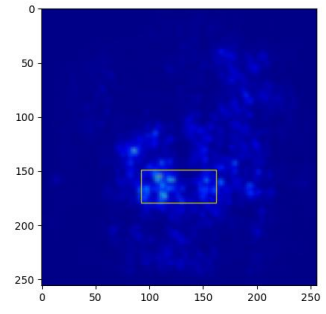
Bounding Box 的選取我們針對三個參數調整:

1. Bbox_Thershold : Saliency map 超過 Bbox_Thershold 值的點才納入 bounding box 計算。

2. Class_Threshold : 分類器預測出某個 Class 的機率超過 Class_Threshold 才針對這個 Class 畫 Saliency map 。

3. k : 在剔除 Saliency map 的 outlier 時，應該 iterate k 次。

我們首先作了以下實驗，以尋找最合適的 k 值。不同 k 值會大幅影響 bounding box 的範圍，太大的 k 會使 bounding box 範圍限縮在影響力最高且最集中的區域，太大的 k 則會喪失去除 outlier 的效果，因為任何高於 threshold 的值都會被納入 bounding box。我們大約對 100 張 validation 的圖片做圖，k=5 是比較合理的參數。而後我們調整了 Class_Threshold 與 Bbox_Thershold，分別在 0.5 與 0.95 時會出現最好的結果

k=1	
k=5	
k=9	

Class_Threshold	Bbox_Thershold	k	score
0.5	0.95	5	0.07291666666666667
0.5	0.95	9	0.07196969696969696
0.5	0.97	5	0.061931818181818185
0.7	0.95	5	0.05416666666666667

5. References:

LCNN: Low-level Feature Embedded CNN for Salient Object Detection
 Deep Residual Learning for Image Recognition
 CNN: Single-label to Multi-label