

基於醫學 X 光影像的病灶點定位

以肺炎為例

第 12 組

F74082264 陳彥博

F74086268 蔡宇曉

F74082010 陳昭穎

一、簡介

在判別病人所患之疾病時，辨認出 X 光影像中的病灶點為一項重要的任務，而這項任務在現今新冠肺炎疫情流行的當下變得更加重要。因此，我們希望能夠利用 region detection 的技術，快速並準確地辨認出影像中可能病灶點，藉此輔助醫師更加方便地作出診斷。

二、動機與目的

動機

1. 我們沒有做過 region detection 相關的主題
2. 主題與課程主旨切合
3. HW2 有提供現成的資料

目的


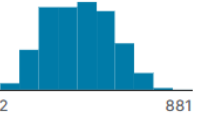
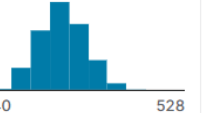
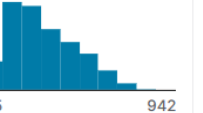

期望能夠在臨床上輔助醫生判斷病人的病灶點，藉此更快、更方便地作出判斷並下診斷。

三、資料集介紹

原先預計使用 HW2 提供的資料集進行實驗，但由於該資料集中被標記為肺炎的資料數量不多，因此較不適合本次實驗。因此，我們在改為使用 Kaggle 上 RSNA (Radiological Society of North America) 提供之胸腔 X 光影像資料集。
([RSNA Pneumonia Detection Challenge | Kaggle](#))

在本資料集中包含了 26684 筆大小為 1024x1024 的胸腔 X 光的影像，並以 Dicom 檔的形式儲存，在 Dicom 檔中也包含額外的病人資訊，而本次實驗中主要會使用胸腔 X 光影像進行模型的訓練。

除此之外，本資料集中也針對有肺炎徵兆的影像提供了對應的 bounding box 資訊，並且以 (x, y, w, h) 的形式表示，(x, y) 為 bounding box 左上角的座標，w, h 分別表示 bounding box 寬與高，而此時 target 欄位的值會是 1。反之若是該影像中沒有肺炎的跡象，則不會標記 bounding box，且 target 會是 0。

patientId	# x	# y	# width	# height	# Target
26684 unique values					
00436515-870c-4b36-a041-de91049b9ab4	264.0	152.0	213.0	379.0	1
00436515-870c-4b36-a041-de91049b9ab4	562.0	152.0	256.0	453.0	1
00569f44-917d-4c86-a842-81832af98c30					0

四、資料前處理

為了使影像能夠符合模型所需之輸入格式，並且擴充資料集讓模型在預測時能有更好的表現，我們對資料進行了以下處理。

1. 從 Dicom 檔中取出具有肺炎的影像
2. 為了提升模型運行效率，將影像從 1024*1024 resize 成 256*256
3. 將灰階數值從 0~1024 修改成 0~256
4. 配合影像縮放調整 Bounding Box 數據
5. 使用 Albumentations 對圖片進行 Transform

```

album.RandomBrightnessContrast(p = 0.5),
album.RandomRotate90(always_apply=True, p=0.5),
album.HorizontalFlip(p=0.5)

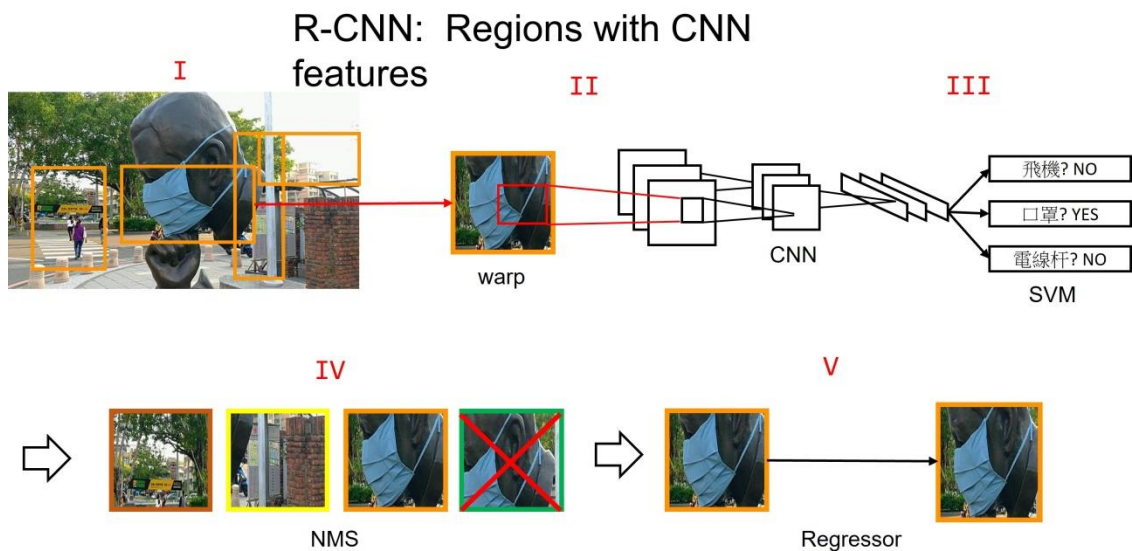
```

在本次期末專題中，我們採用之訓練集大小為 4806，驗證集與測試集大小則分別為 601 與 602。

五、模型介紹

在 region detection 的領域中，RCNN 及其延伸模型為最常使用的模型。底下將會對不同模型進行介紹。

RCNN



演算法流程

1. 生成候選區域：使用 Selective Search 產生大約 2000 個候選區域 (Region proposal)
2. 獲取區域特徵：將每一個候選區域經過裁切或變形後(Crop/ Warp) 後丟入 CNN 中得到固定維度的輸出(CNN feature)
3. 判斷區域類型：將 CNN 結果丟入分辨各種類別的不同 SVM 中進行評分分類
4. 刪除多餘區域：使用 NMS 將過於重疊的候選區域刪除
5. 精修候選區域：最後再由回歸模型修正 Bounding Box Position

Selective Search

Selective Search 的理論為使用圖像分割 (segmentation)，並套用階層群聚演算法(hierarchical grouping algorithm)，藉此產生物體的候選區域(object proposal)，最後再使用 SVM 辨識物體。

而現實中的物體是多尺度 (Multiscale) 的，沒有辦法使用任何單一尺度來涵蓋所有物體，需要使用不同的特徵並考慮不同大小的區域才能達成。如下圖所示：



(a)



(b)



(c)



(d)

- a) 碗和湯匙表現出一種階層式的結構
- b) 兩隻貓咪顏色不同但紋理相同
- c) 變色龍和樹葉顏色相同但紋理不同
- d) 車殼和輪胎顏色紋理皆不同，但其實為同一物體(車子)

階層群聚演算法

根據各區域的相似度依序合併產生最後的候選區域，考慮的相似度類型包含但不僅限於顏色、紋理、大小、形狀等等

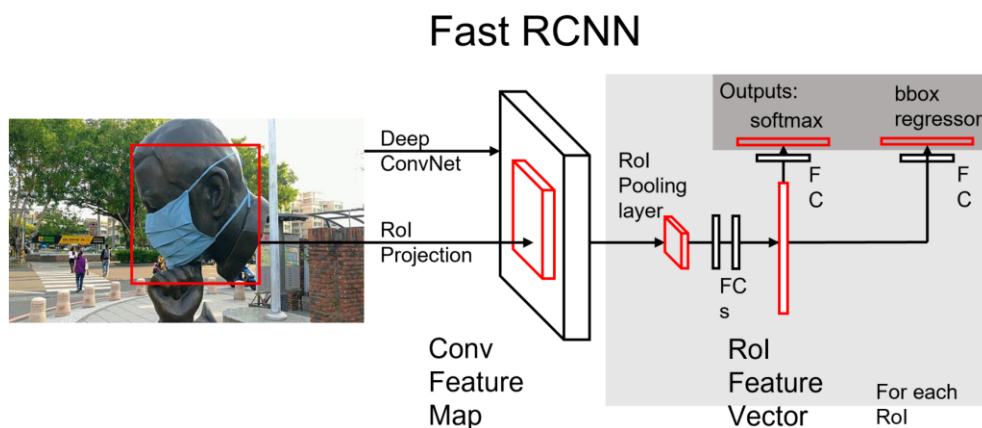
NMS(Non-Maximum Suppression，非極大值抑制)

1. 將所有 boxes 依照 SVM 所預測出的分數由高到低排列
2. 依序遍歷每一候選區域，如果有其他區域和當前所選中者 IoU 大於閾值，就將其刪除(代表其他區域和此區域有高相似度，因此留下得分高者)
3. 持續進行直到所有區域處理完畢

RCNN 之缺點

1. RCNN 一開始必須產生約 2000 個 region proposal，且每個區域都要放入 CNN 提取特徵，因此有很多重複運算的部分非常花時間
2. 在將候選區域放入 CNN 前需要先將候選區域重新調整尺寸至 227×227
3. CNN 提取特徵的層數較高，因此候選區域對於小區域較不敏感
4. 需要對 bounding box 訓練線性回歸模型以便回歸物體精確的位置

Fast RCNN



Fast R-CNN 會將整張影像放入 CNN 中計算 Feature Map，再將候選區域投射到 Feature Map 上，並利用 RoI Pooling 計算出 RoI Feature Map。

由於候選區域極有可能會互相重疊，因此此方法減少了 RCNN 將每個候選區域分別放入 CNN 中所產生的重複計算，藉此提高運算效率。

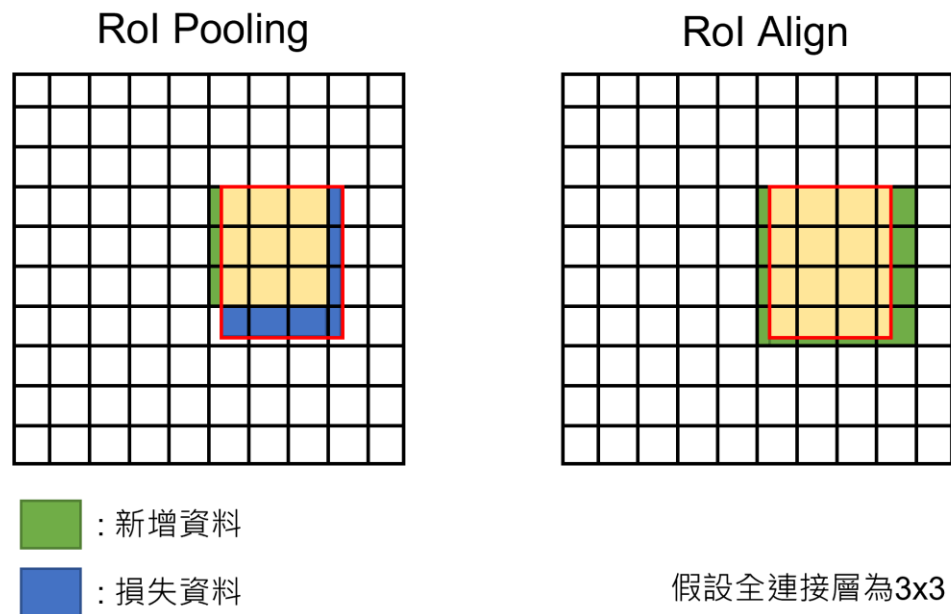
Fast RCNN 之缺點

在進行 RoI Pooling 的過程中，因為候選區域的座標通常不會是整數，因此會先進行一次取整數才能對應到 Feature Map 上，然而這個過程會損失一些資訊（被捨棄的小數點以後）。而在放入後續的全連接層時，候選區域和全連接層的大小可能無法整除匹配，這時又需要捨棄某些資訊以繼續進行。而這個過程會導致模型遺失了許多候選區域的資訊。

解決方法 — RoI Align

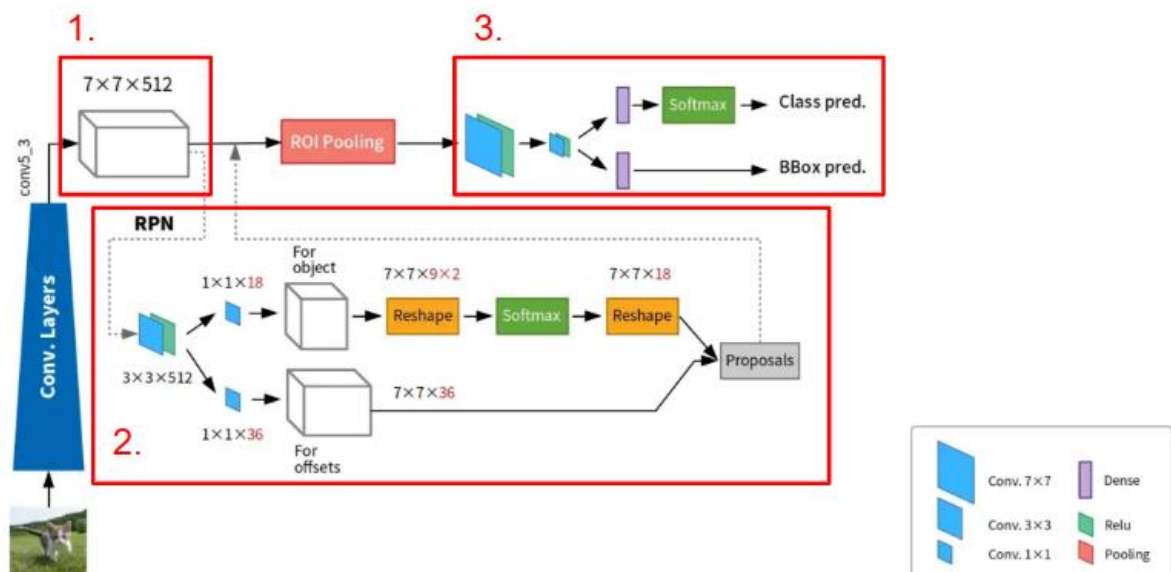
類似於重新設立座標軸，將對應原始候選區域的 Feature Map 直接劃分縮放成能匹配後續全連接層的大小，因為新的單位區域包含數個不同的舊單位區域，因此需要透過差值運算與比例分配決定每一個新單位區域的資訊，透過此方法便

能不改變及遺失原始候選區域的資訊。



Faster RCNN

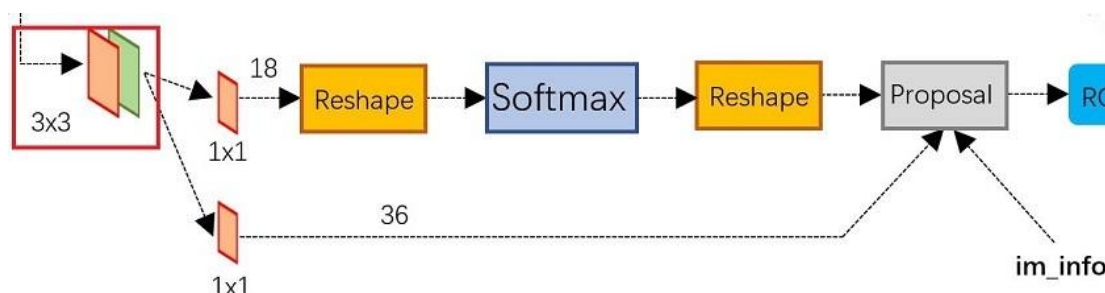
本次實驗中採用 Faster-RCNN 模型進行訓練，其運作流程如下所示：



1. CNN → 產生影像 Feature Map
2. Region Proposal Network → 生成並修正候選區域
3. FCNN → 為候選區域評分與分類

Faster RCNN 拋棄過往透過 selective search 來獲得候選區域的方式，改成在 CNN 生成全圖的 Feature Map 後利用 RPN(Region Proposal Network) 進行生成，藉此大幅減少生成候選區域所需的時間。

RPN(Region Proposal Network)



RPN 會先從 Feature Map 中透過固定演算法產生許多 Anchor Box，而後上方架構主要將 Anchor Box 分成 Positive 和 Negative 兩類，下方架構則是針對 Anchor Box 計算偏移量，而最後的 Proposal Layer 會將 Positive Anchor 和偏移量結合成最終的候選區域。

六、衡量標準

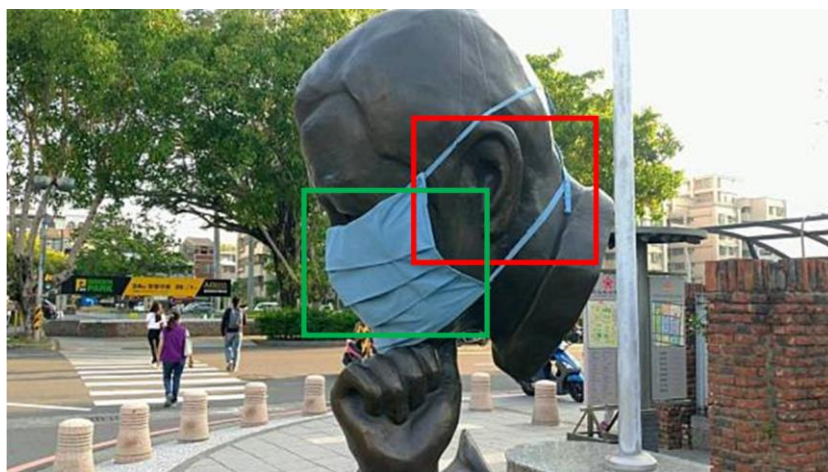
在評斷模型表現好壞時，我們採用了兩種衡量標準：

1. IoU (Intersection over Union)

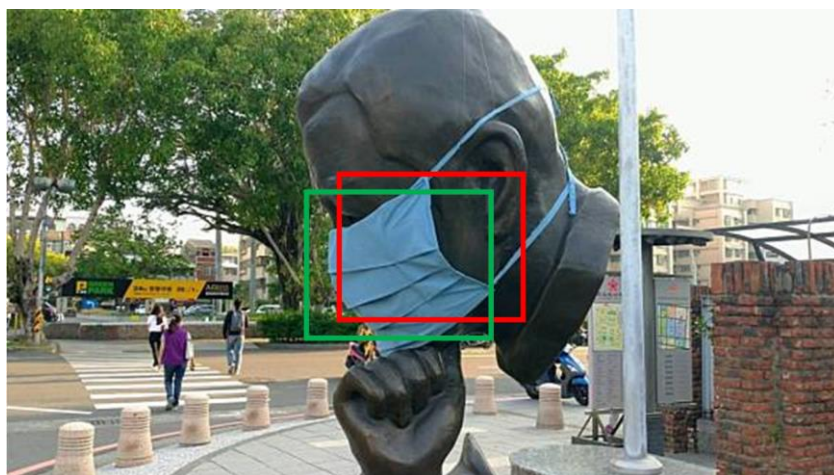
藉由計算 label 標記的 bounding box 區域與模型標記的 bounding box 之重合狀況來表示模型預測結果的好壞。其公式如下所示

$$\text{IoU} = \frac{\text{DetectionResult} \cap \text{Groundtruth}}{\text{DetectionResult} \cup \text{Groundtruth}}$$

當重疊區域低，IoU 值較不理想(0.15)



而重疊區域高，會有較高的 IoU 值(0.69)



由於許多影像中會有多個 bounding box 存在，因此我們原本在計算 IoU 時會：

1. 收集模型預測之 confidence score 超過某 threshold 的 bounding boxes
2. 對每個 label 找出上述 bounding boxes 中重合度最高者進行 IoU 運算
3. 計算平均

但我們發現這樣並不合理，因為實際場域中模型不會提前知道影像中有幾個病灶點(labels)

因此我們改採用以下計算方式：

1. 收集模型預測之 confidence score 超過某 threshold 的 bounding boxes
2. 使用所有 labels 與上述 bounding boxes 計算 IoU，而非一一對應的關係

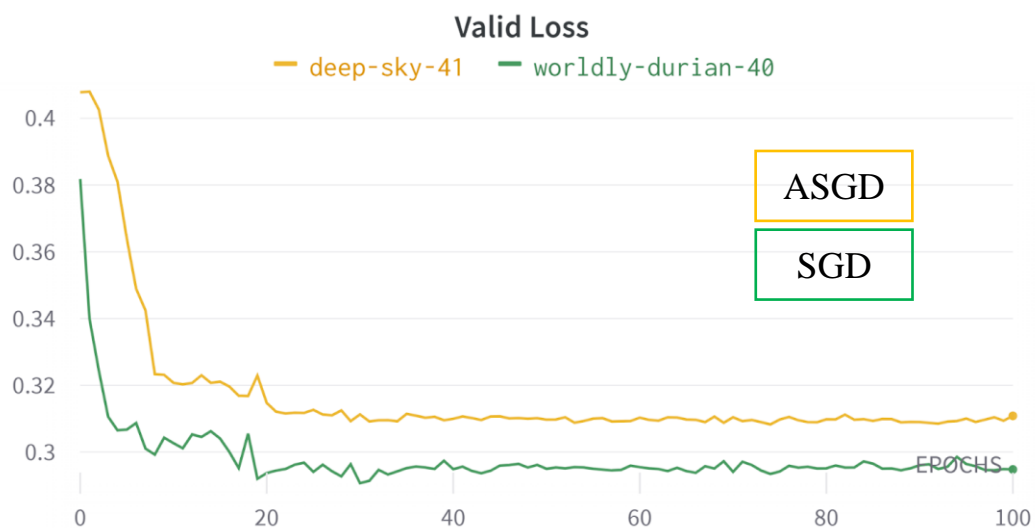
2. Dice coefficient

也可以稱作 F1-score，是機器學習中常用的評量指標，其公式如下所示：

$$\text{DiceScore} = \frac{2 * TP}{2 * TP + FP + FN}$$

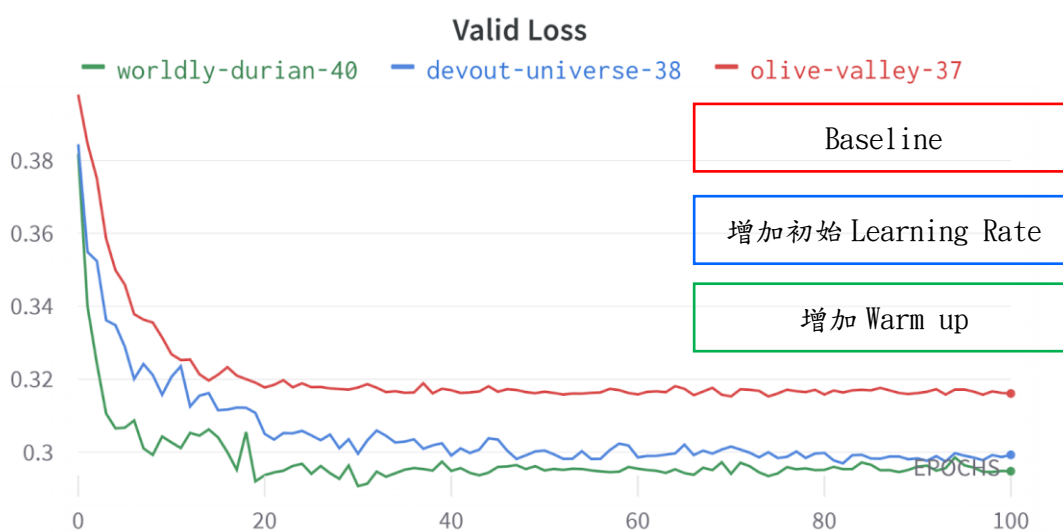
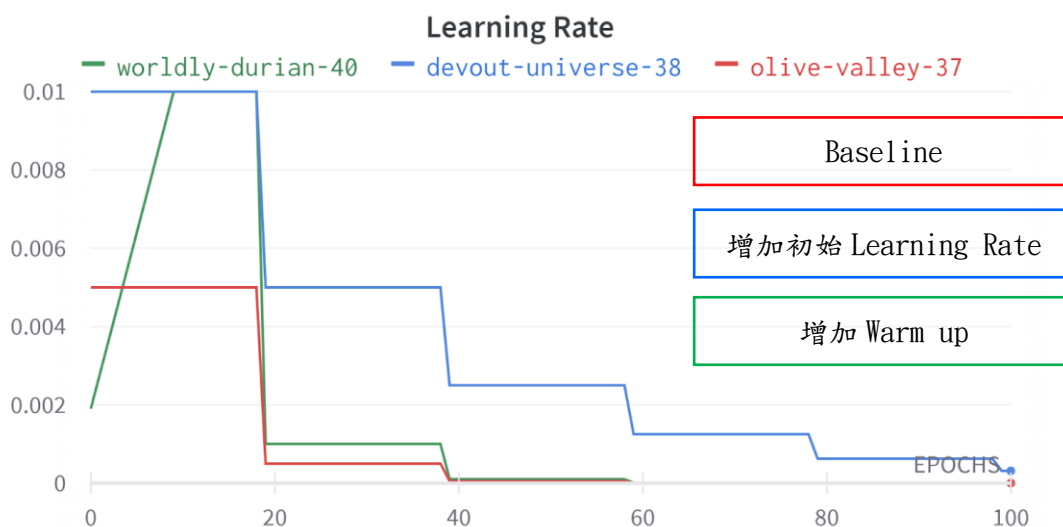
七、實驗設定與結果分析

參數調整 — Optimizer



由上圖可知，SGD 的表現比 ASGD 佳，而使用 ADAM 的模型之參數已無法更新。

參數調整 — LR Scheduler



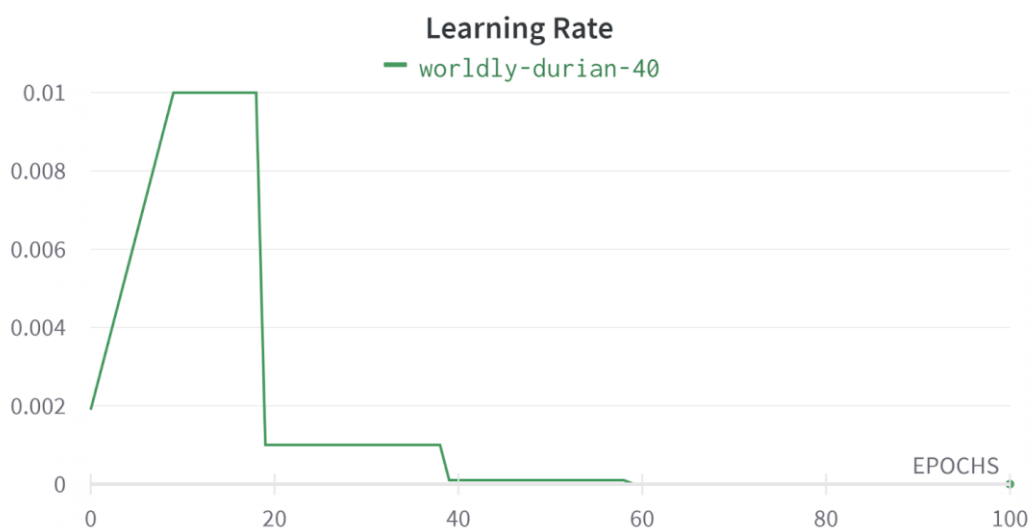
由上圖可知，在提高初始 Learning Rate 後，模型的表現已經有所提升；而在增加 warm up 的設定後，模型表現又進一步地提升。

參數調整 — Others

- Epochs: 配合 Learning Rate 調整模型收斂程度
- Batch Size: 調節模型參數更新率與整體訓練時間

實驗結果

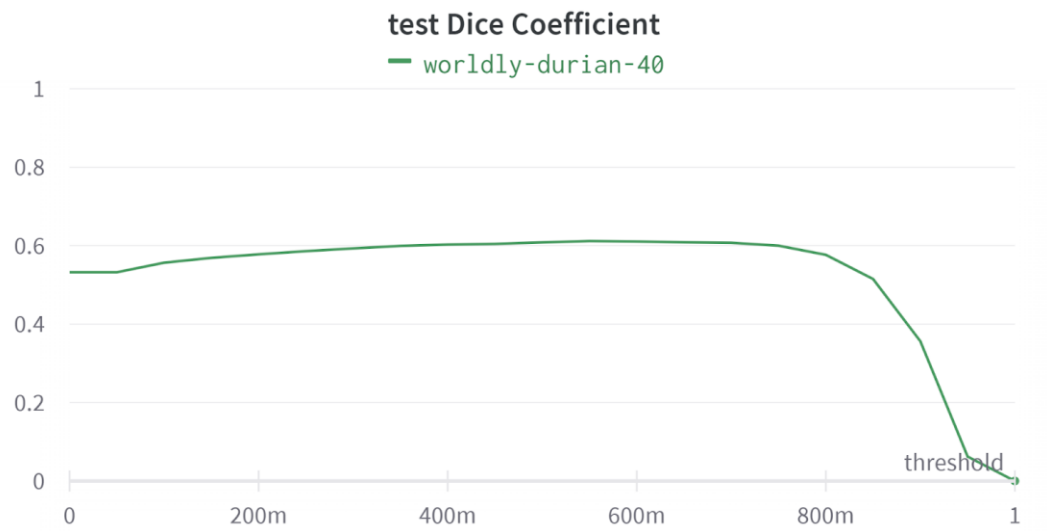
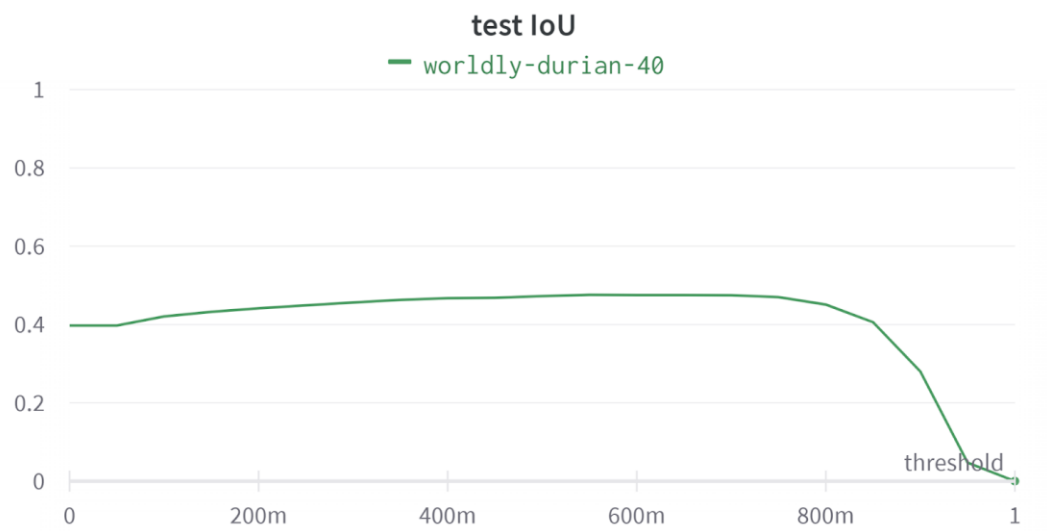
Hyperparameters	Value
Loss Function	CrossEntropy
Learning Rate	0.01
Training Epochs	100
Optimizer	SGD
Threshold	0.55
LR Scheduler	線性上升+ 階層式下降



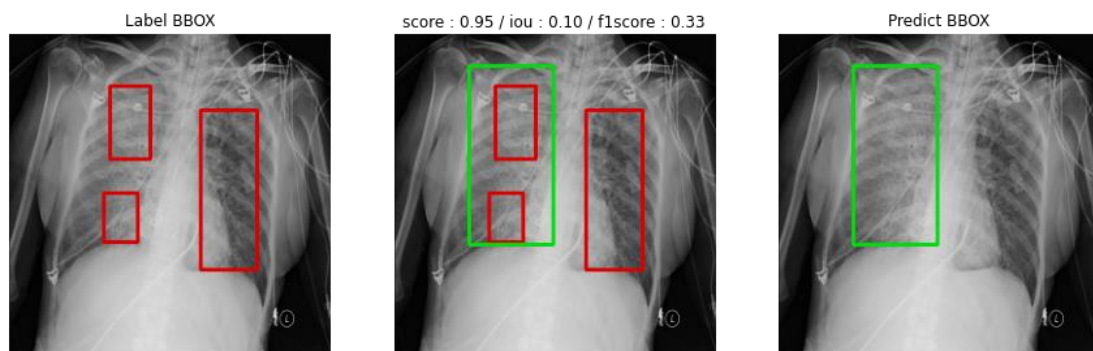
上表為我們在實驗中使模型表現最好之設定，其 Learning rate 走向也如上圖所示。

而下方之表格與圖片呈現了在該設定底下，模型在預測 test data 之 bounding box 時的表現。

Metrics	Value
IoU	0.4760
Score	0.8158
Dice-Coefficient	0.6118

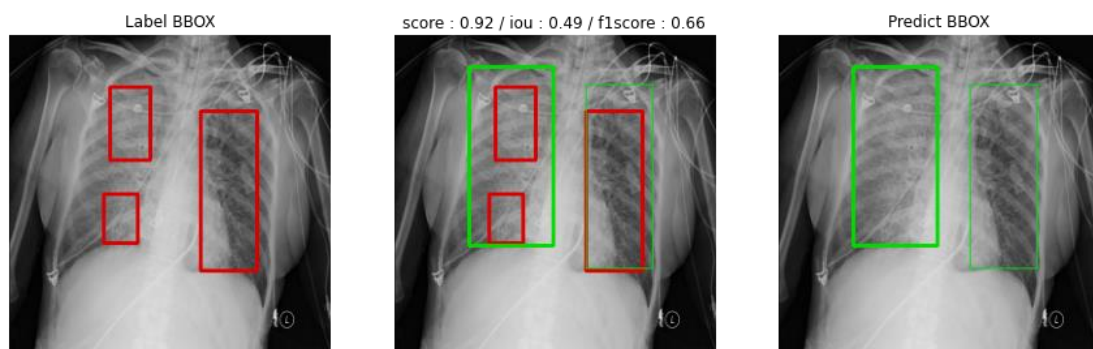


實驗結果 — 不同 IoU 計算方式對照



修正前：

- IoU:0.10
- F1Score:0.33



修正後：

- IoU:0.49
- F1Score:0.66