**肺腺癌病理切片影像之腫瘤氣道擴散偵測競賽 II：運用影像分割作法於切割STAS輪廓**

**組別: Team 1377**

1. 環境

作業系統: google colab

語言: Python 3.7.13

套件(函式庫):

numpy 1.21.6

sklearn 1.0.2

Pillow 7.1.2

opencv-python 4.1.2.30

matplotlib 3.2.2

torch 1.11.0

segmentation-models-pytorch 0.2.1

albumentations 0.1.12

預訓練模型:

Feature Pyramid Network (FPN) in segmentation-models-pytorch

Pretrained encoder: efficientnet-b3

額外資料集: 無

1. 演算方法與模型架構

說明演算法設計、模型架構與模型參數，包括可能使用的特殊處理方式。

我們的演算法總共分三個步驟: 前處理、模型訓練、後處理。

在前處理的部分，首先我們把競賽所附訓練資料的json檔，轉換成訓練模型所需的mask圖片。接著在將原始訓練資料圖片與產生的mask圖片丟入模型訓練前，會經過三個步驟: 第一為把上述圖片調整為800x800像素大小。接著使用Argumentation的方法，增加訓練資料集的多樣性，使用的圖片轉換方法有: HorizontalFlip、Rotate、VerticalFlip、ShiftScaleRotate、HueSaturationValue、Sharpen、RandomBrightnessContrast、Crop、PadIfNeeded。最後利用segmentation-models-pytorch套件的函式get\_preprocessing\_fn，將encoder名稱efficientnet-b3及imagenet的參數值傳入，得到一個預處理的函式，將圖片通過此預處理函式及albumentations套件裡的函式lambda轉換後，即傳入模型訓練。

在模型訓練的部分，模型架構使用的是segmentation-models-pytorch套件裡的Feature Pyramid Network (FPN)模型，而FPN中使用到的encoder是已經預訓練好的encoder efficientnet-b3。模型參數值來源為imagenet，參數量有10M。模型所使用的activation function為sigmoid，optimizer為torch套件裡的Adam，初始的learning rate設為0.0001。模型訓練的過程總共會經過32個epoch，每一個epoch更新參數所需的訓練資料量batch size設為4。使用的模型表現的衡量標準中，accuracy為segmentation-models-pytorch套件裡的Fscore，loss為segmentation-models-pytorch套件裡的DiceLoss。

在後處理的地方，我們將競賽的測試資料傳入，所得到模型輸出結果為800x800像素大小的mask圖片，其中STAS的部分是紅色的，非STAS的部分為黑色。我們將輸出結果圖片的STAS部分轉成白色，並把圖片調整大小至原測試圖片大小。

1. 資料處理

首先，我們將datasets中提供的.json檔案，透過檔案中的資料得到每一張datasets對應的mask。參考的pseudo-code如下：

anno is a list. // anno is a list of “.json” file.

for jsonfile in len(anno) do

anno\_file = (path of jsonfile)

file = openRead(anno\_file)

data = json.load(file) // 將檔案中的資料指派給data

mask = np.zeros((data['imageHeight'], data['imageWidth'], 1))

//

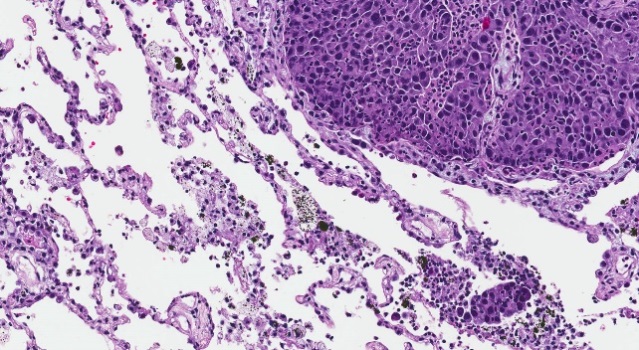
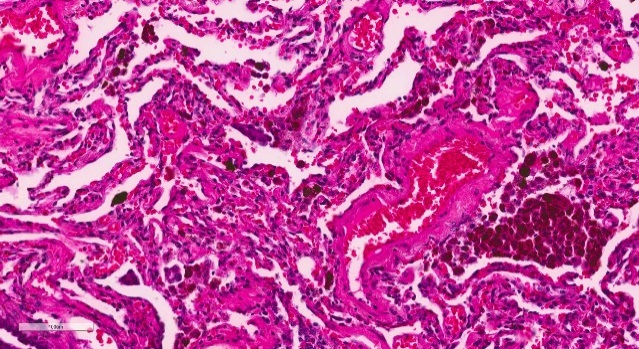
Convert mask data to image and let result be equal to mask.

Finally, store picture into the assigned directory.

//

end for

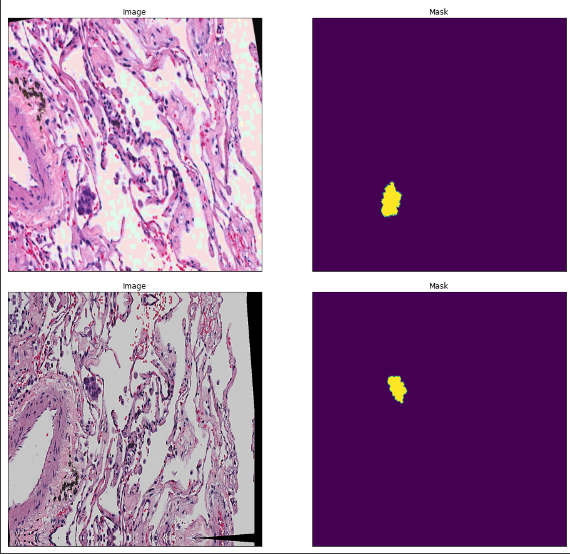
接著，我們利用sklearn.model\_selection當中的train\_test\_split()隨機產生訓練以及測試的train datasets與valid datasets，並沒有特別進行資料的刪減或增補。至於train\_test\_split中的參數設定，我們將test\_size設定為0.3，也就是說引入一個dataset，train與valid datasets的資料數量比為7:3，並且給random\_state值隨機分割資料。

除此之外，我們發現到在公開資料集中，圖片的顏色有些微的差別。

圖一、主辦方提供資料集之其中兩圖

以上面兩張為例，圖為資料集中的Public\_00000073.jpg（左）與Public\_00000064.jpg（右），可以發現左圖明顯偏紅色，右圖明顯偏紫色。由此可知，在處理這些資料時，除了需要考慮到肺腺癌病理切片本身的特徵（例如細胞數量較其他部分異常地多），也需要考慮會有圖片顏色、明暗、對比度等問題，若引入資料處理不當，可能會影響訓練效果。故我們除了使圖片旋轉，也設法改變圖片的亮度與對比度。在此專案，我們也有做圖片基本的rotate、flip以及scale。於此，我們使用Albumentationsc函式庫，使得我們可以調整圖片的對比、明暗。

最後，可得到調整過後的圖片與mask如下：



圖二、經由Albumentationsc處理過後的圖片

1. 訓練過程

在訓練模型前，我們使用torch.utils.data中Dataloader函式，設定batch size、workers number等相關參數。參數說明如下：

1. pin\_memory：在回傳之前，複製Tensor到使用cuda的記憶空間。
2. batch\_size：每一個batch需要載入的取樣數量大小。
3. shuffle：讓資料集重新排列。
4. num\_workers：在載入資料時，使用subprocesses的數量。

接著，我們需要計算模型的loss。而因為此次競賽的評分方式以dice loss結果進行評分，故引入segmentation\_models\_pytorch函式庫，並使用其中的Diceloss()函式計算整個模型的loss。最後，我們也需要設定Adam optimizer的參數。

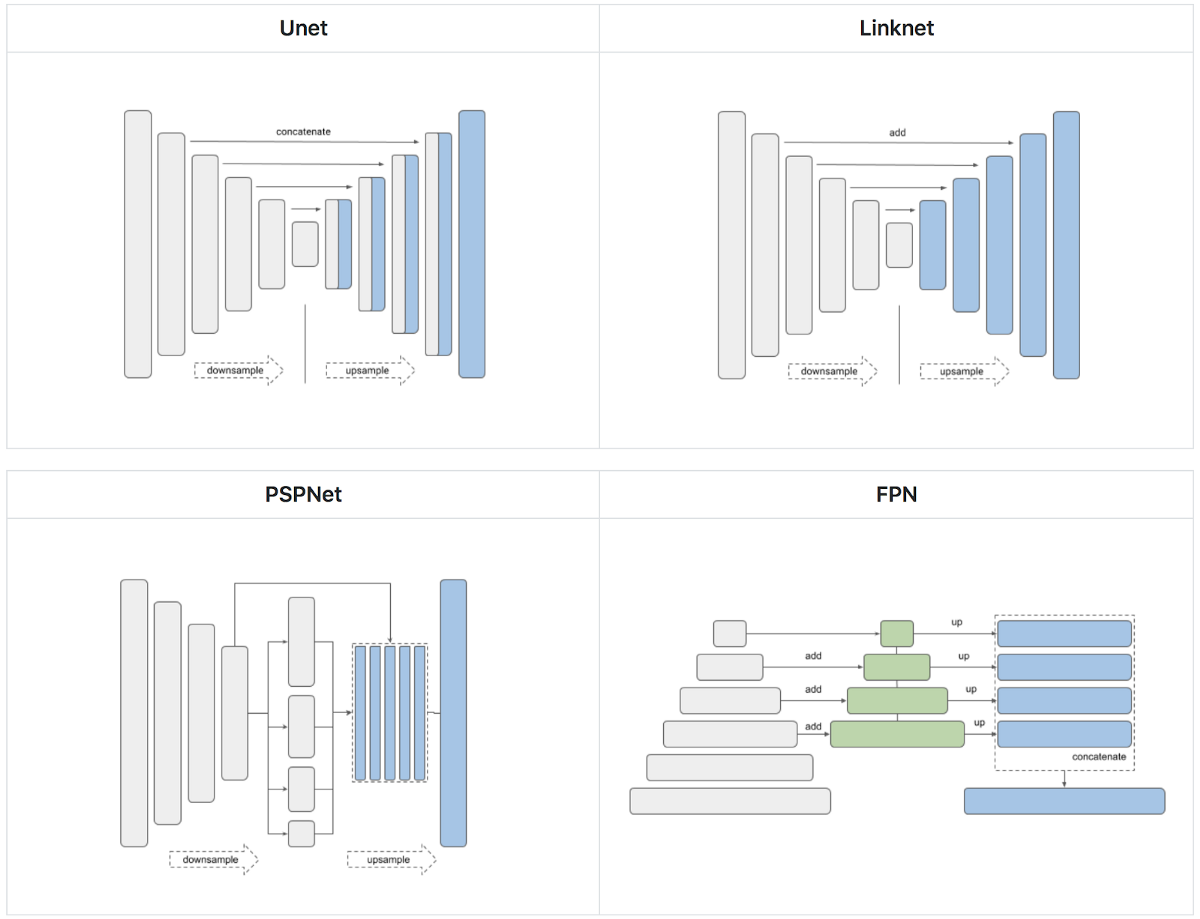
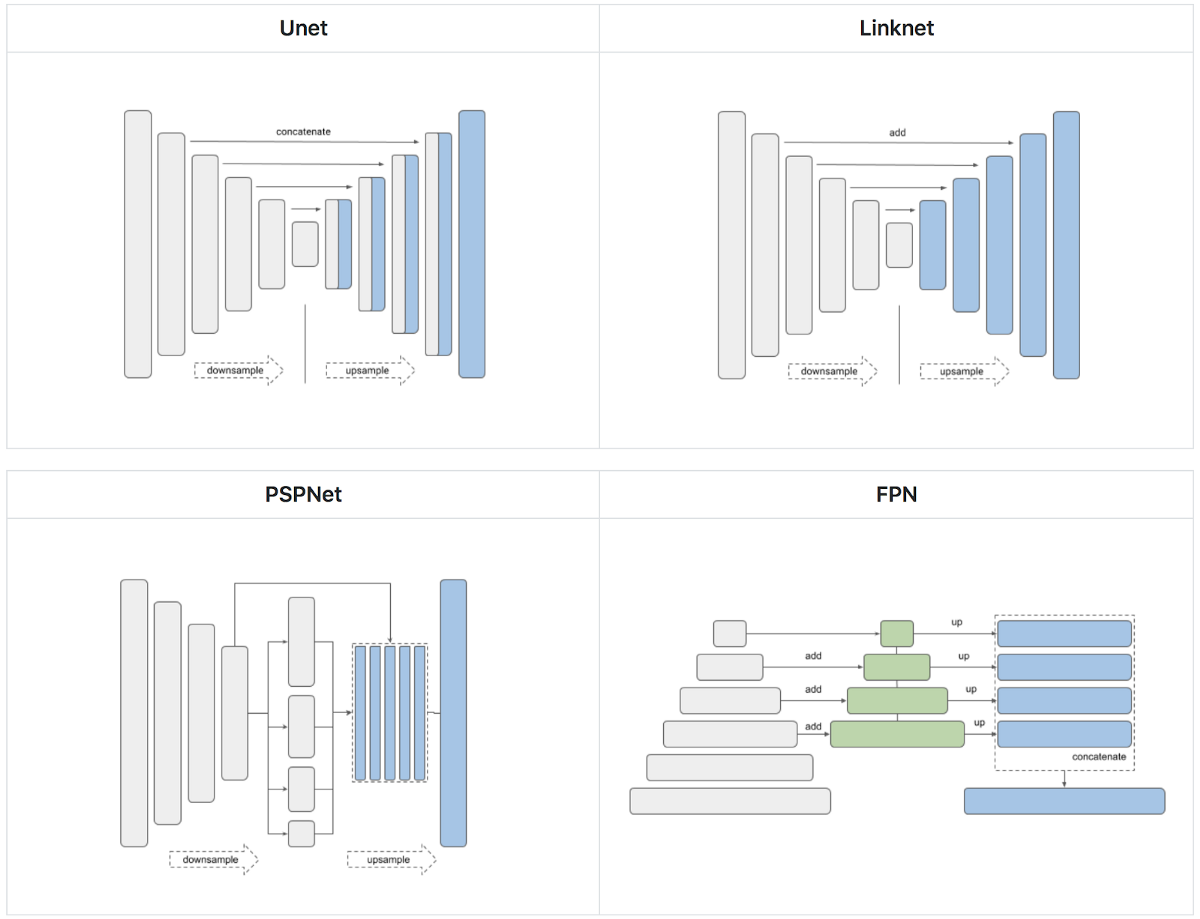
這些前置作業設定完畢後，將epochs設定為32，讓模型反覆訓練。在訓練過程中，若產生目前最佳結果（score值最大），就會儲存該次訓練產生的模型、權重（weight）。並且於訓練完後，將validation數據之score最高時產生的weight數據，也是我們的訓練結果，儲存在雲端位置，方便後續進行測試使用。

以下總結設定參數與訓練模型相關資料：

|  |  |
| --- | --- |
| Model | best\_mode\_FPN\_efficientnet-b3\_aug.pth |
| Backbone | FPN(Feature pyramid network) |
| Classes | [“bg”, “stas”] |
| Learning rate | 0.001 |
| Epochs | 32 |
| Batch size | 4 |
| Workers number | 16 |
| Score threshold | 0.5 |

1. 分析與結論

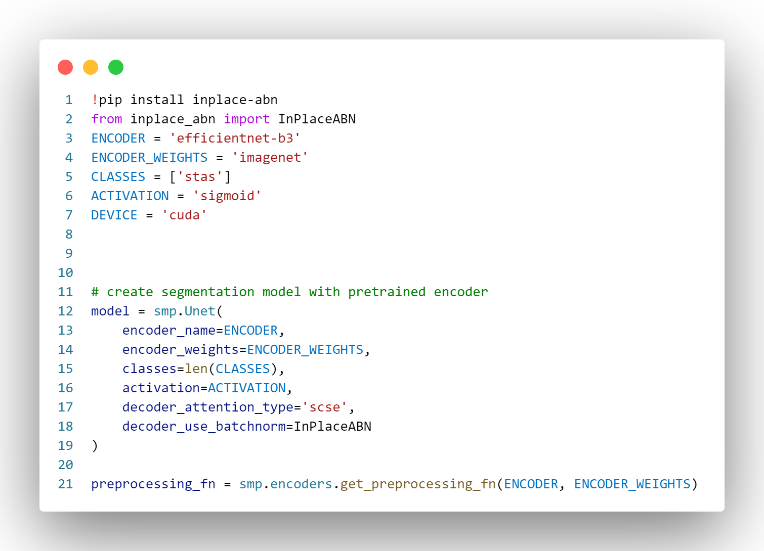
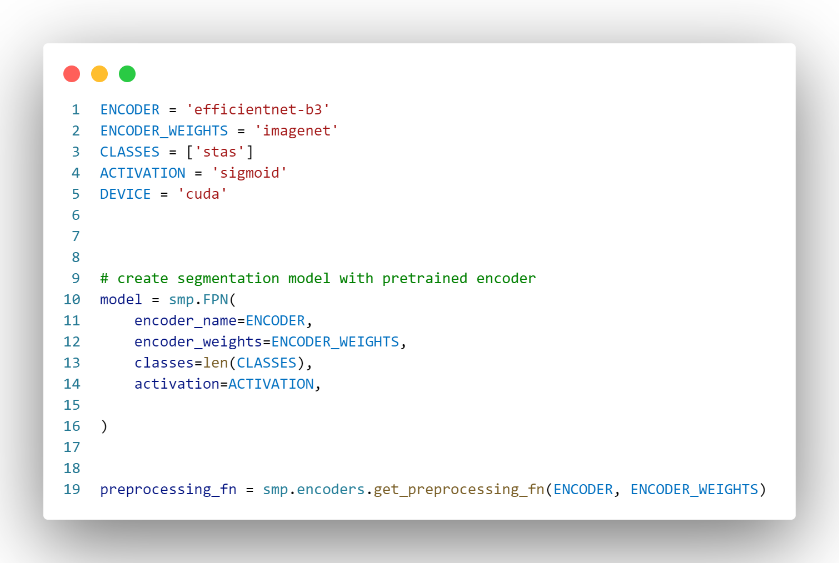
這次我們主要比對的模型是FPN(Feature pyramid network)跟U-Net。U-Net與FPN最大的差別就在FPN會在不同層的特徵上做預測，而U-Net只拿最後一層特徵，不過在網路的結構上是相似的。



圖一、U-Net vs. FPN

FPN的發起點在於做物件偵測的多尺度問題，即如何同時擁有大、中、小物件的高準確度的偵測結果。實際上FPN使用簡潔的方法，同樣有up-/down-sampling跟skip connection，但將淺層的特徵用更小的convolution (1x1)加到深層的特徵。並且在up-sampling的過程中，對每張特徵張量都做預測。

針對這兩個model，我們同樣使用efficientnet-b3作為encoder，使用imagenet作為encoder weights，以及sigmoid作為activation function。另外，U-Net加入attention block來提高準確率，以及使用In-place activated batchnorm來降低記憶體使用量。

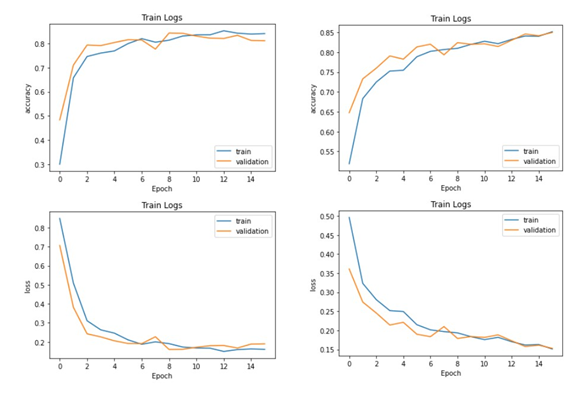


圖二、Model參數

由於FPN是出自detection任務的，我們原先認為U-Net在這次的task會表現更好，在16個epoch時U-Net在public dataset表現的確是比FPN要好的。但分析training loss的趨勢可以發現，FPN的accuracy還有上升的空間-我們決定加長訓練到32個epoch。期間我們也有替換optimizer (Rectified Adam, team-1377\_v4)做測試，對於learning rate比較不敏感，但結果變化不大。

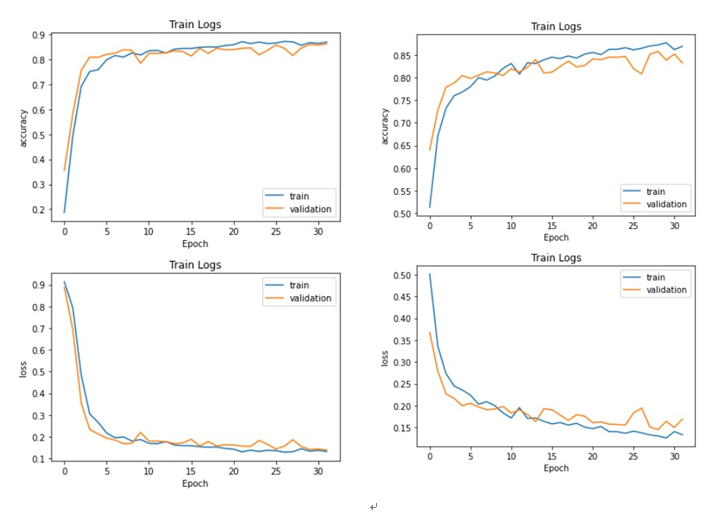


圖三、Submission History



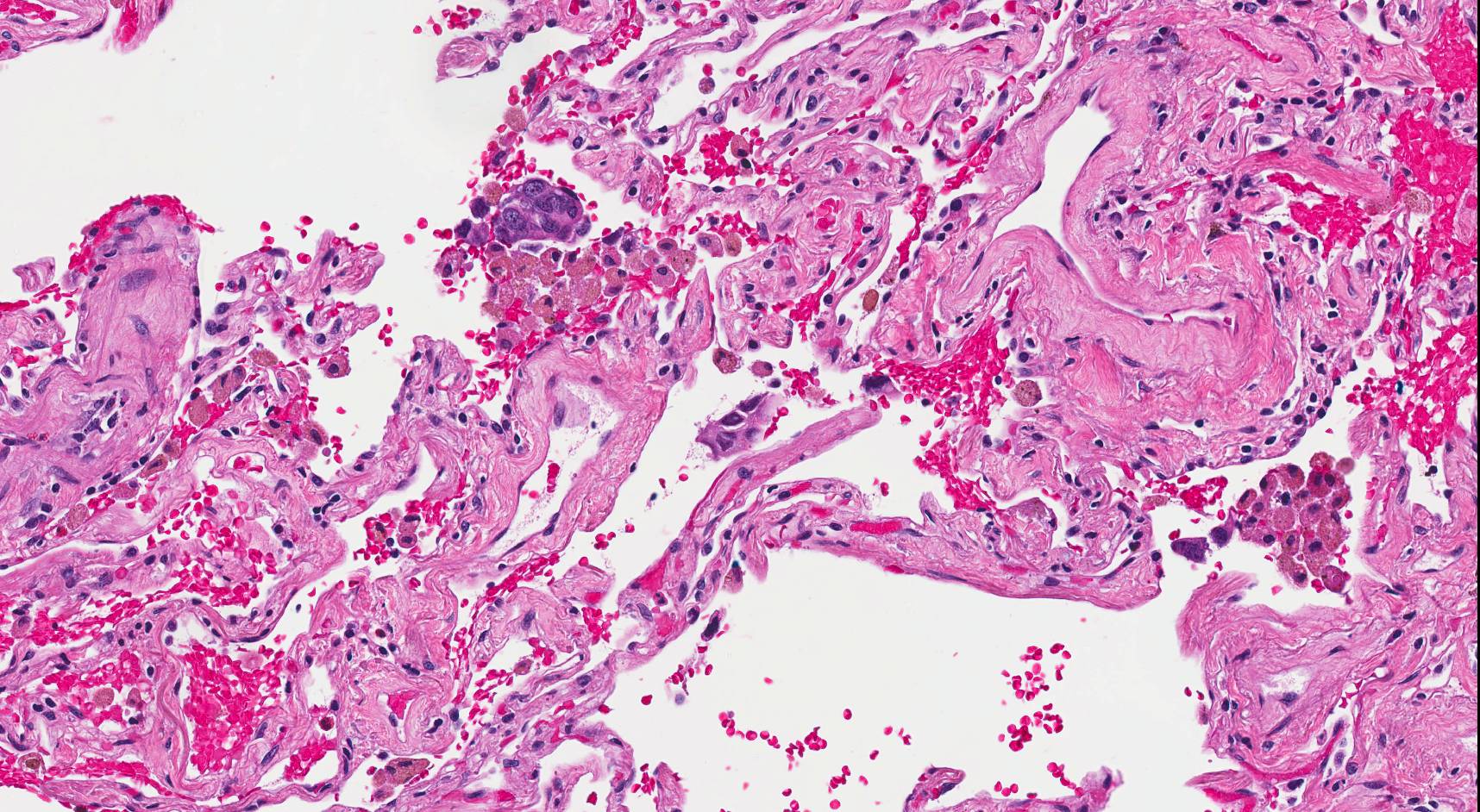
圖四、Train logs for EPOCH=16 (Left: U-Net, Right: FPN)

在訓練到第32個epoch的時候，U-Net的accuracy看起來更高，但是在public跟private dataset的表現都有所下降，可能是發生overfitting了，但是FPN的public分數有明顯的提升。這是我們的最佳結果。

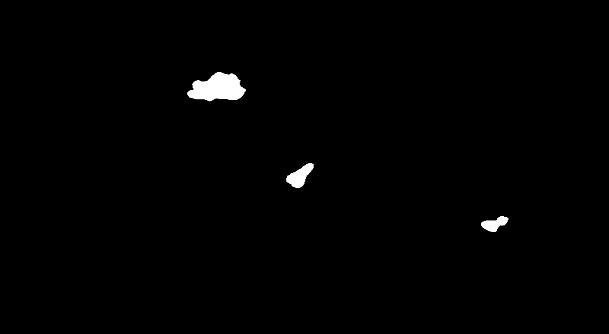
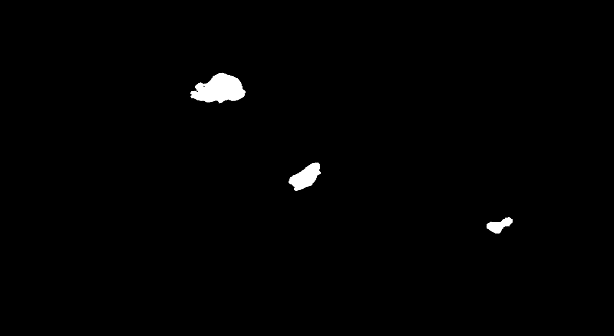


圖五、Train logs for EPOCH=32 (Left: U-Net, Right: FPN)

我們取兩個model的最佳結果來測試(U-Net, 16 epochs / FPN, 32 epochs)，可以看出U-Net在segmentation上有遺落的部分，FPN則是有完整分割出STAS的區塊。



圖六、Test Image

圖七、Output Mask (Left: U-Net, Right: FPN)

未來改進，會朝向在post processing移除過小的mask component (pixel數<threshold)，以及結合不同model的結果來做model ensemble。另外也可能嘗試gradient accumulation分割成mini batch來測試不同batch size的影響

1. 程式碼

◎由於權重的保存是以最佳的訓練結果，故在private data得到的最高分數的模型權重，沒有留存下來

-可執行性

1.使用google colab環境進行訓練，硬體資源配置如下:

GPU RAM : 15.11 GB

GPU Type : Tesla T4

NVIDIA-SMI : 460.32.03

Driver Version : 460.32.03

CUDA Version : 11.2

因此使用的硬體資源為可接受範圍，故程式對硬體資源要求不算苛刻

2.訓練時間為 30min/10epochs ，總共訓練32個epochs，故總訓練時間約為100分鐘

-可驗證性

由於讓模型有更大的隨機性，故沒有對參數進行初始化以及RANDOM SEED的設定，但是使用同一組參數，如下:

Optimizer : ADAM

Learning Rate : 0.0001

Batch Size : 4

Epochs : 32

Learning Rate Scheduling: N/A

每次訓練結果f1score皆落在 0.86 ，因此程式的可驗證性是高的

1. 參考資料

1.Xiaojing Qiu(2022). U-Net-ASPP: U-Net based on atrous spatial pyramid pooling model for medical image segmentation in COVID-19.

2. Aleksandar Vakanski , Min Xian , and Phoebe E. Freer(2019). Attention-Enriched Deep Learning Model for Breast Tumor Segmentation in Ultrasound Images. arXiv, 1910.08978.

3. Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015).U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv, 1505.04597.

4. Tsung-Yi Lin, Piotr Dollar , Ross Girshick , Kaiming He , Bharath Hariharan , and Serge Belongie (2017). Feature Pyramid Networks for Object Detection. IEEE Xplore.

5. Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder(2017). In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. arXiv, 1712.02616

1. 聯絡資料

* 隊伍

|  |  |  |
| --- | --- | --- |
| 隊伍名稱 | Private leaderboard成績 | Private leaderboard名次 |
| Team 1377 | 0.878244 | 43 |

* 隊員

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 學校系所 | 電話 | e-mail |
| 郭哲睿 | 成功大學 電機工程學系 | 0978660326 | louiskuo888@gmail.com |
| 許郁翎 | 成功大學 資訊工程學系 | 0909925337 | sillosougri270415@gmail.com |

* 指導教授

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教授姓名 | 課程 | 課號 | 學校系所 | e-mail |
| 詹寶珠 | 人工智慧導論與實作 | E2-094 | 成功大學電機系 | pcchung@ee.ncku.edu.tw |