資料科學導論
期末專題報告

SIIM-FISABIO-RSNA COVID-19 Detection
(Identify and localize COVID-19 abnormalities
on chest radiographs)

Group3 Member :
李侑諭 涂崇仁 黃子軒 吳尚明

目錄:

Chapter 1

研究背景與動機

新冠肺炎是近年社會一直關注的議題，此類的主題及技術運用也因而增加許多，因此我們組選擇以肺炎的相關主題進行研究，並挑選了這個COVID-19 Detection的主題。

在此主題中，會針對肺部的X光照片的不透明部分進行標記，這對於判斷是否為新冠肺炎是非常重要的依據，訓練目標是給定大量以標記的資料訓練出能夠針對肺部的不透明部分進行分類標記的模型。

在這個專題中，我們的目標是以四人的能力完成一份獨特的作品，並利用在課堂所學在研究的過程中了解資料科學在實際模型訓練運行的整體流程，從資料的蒐集、前處理，到模型的訓練及優化，從自己的角度出發，著手於研究相關的主題，在隊友的相互討論中完成屬於自己的努力結晶，雖然說已目前的能力無法到在社會、醫療院所上有所作為，但多少能夠盡一份心力做貢獻 同時也能開拓視野、關心時事、投入在社會之中。

從我們的角度出發，不被別人成功的方法所拘束

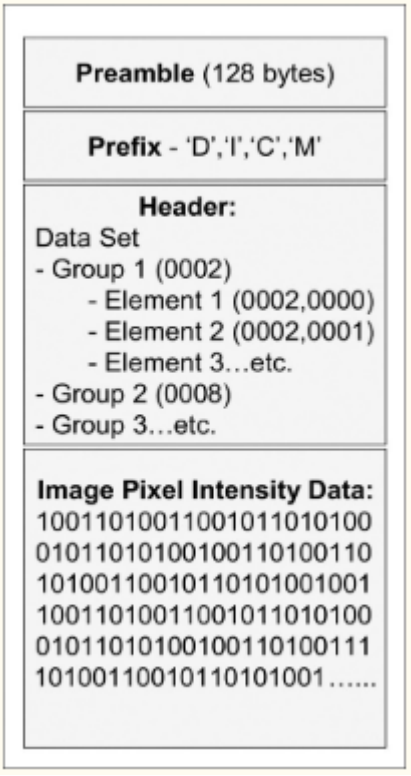
Chapter 2

資料集分析

2.1 載入所有資料集

觀察資料集的檔案，除了兩個標記的csv檔外，訓練的資料主要來自訓練集的dcm檔，裡面不只包含了X光照片更是紀錄許多醫療紀錄，然而要解析DCM檔的資訊就必須先了解DICOM協定。

DICOM即為醫療數位影像傳輸協定，是對於醫學影像的處理、儲存、傳輸的通用協定，藉由傳遞DICOM格式的檔案，接收與交換影像以及病人資料，DICOM影像中包含兩種資料，為像素資料(pixel data)及影像屬性(attribute)，是以TCP/IP為基礎的應用協定，並以TCP/IP聯繫各個系統(下左圖)，將dcm檔的資訊印出，除了像素資料以外也記錄影像的屬性病患的基本資料(下右圖)。

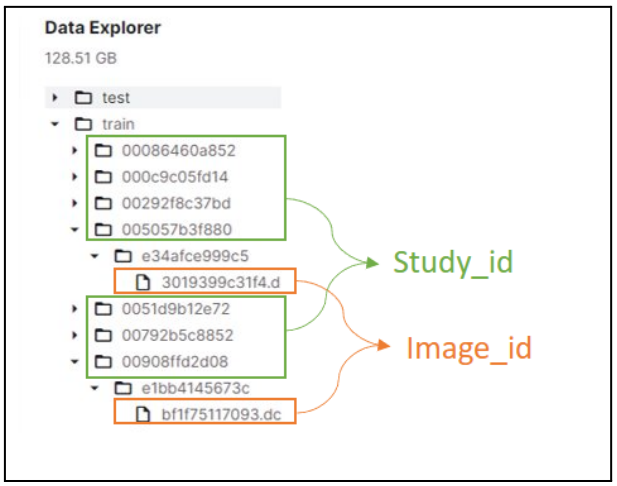



```
sample = dicom.dcmread("/content/51759b5579bc.dcm")
pprint(sample)
```

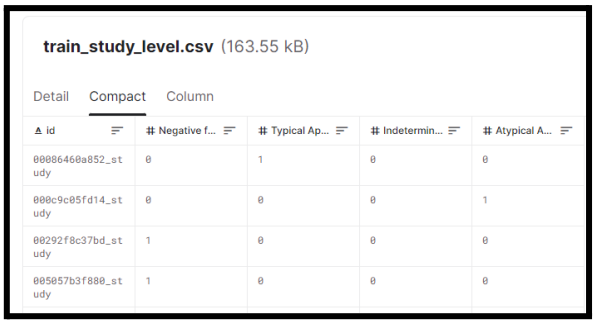
```
(0008, 0005) Specific Character Set              CS: 'ISO_IR 100'
(0008, 0008) Image Type                          CS: ['DERIVED', 'PRIMARY', '
(0008, 0016) SOP Class UID                       UI: 71228e4340de
(0008, 0018) SOP Instance UID                    UI: 51759b5579bc
(0008, 0020) Study Date                          DA: '96fc21dd2b1f'
(0008, 0030) Study Time                          TM: '13e700cac7f0'
(0008, 0050) Accession Number                    SH: '0dc10cf540cf'
(0008, 0060) Modality                            CS: 'DX'
(0009, 0010) Private Creator                     LO: 'GEIIS'
(0010, 0010) Patient's Name                      PN: '3a0c965d2601'
(0010, 0020) Patient ID                          LO: '2c00dc1ead80'
(0010, 0040) Patient's Sex                       CS: 'M'
(0012, 0063) De-identification Method            LO: 'CTP Default:  based on
(0012, 0064)  De-identification Method Code Sequence  6 item(s) ----
   (0008, 0100) Code Value                           SH: '113100'
   (0008, 0102) Coding Scheme Designator             SH: 'DCM'
   (0008, 0104) Code Meaning                         LO: 'Basic Application Co
   ---------
   (0008, 0100) Code Value                           SH: '113105'
   (0008, 0102) Coding Scheme Designator             SH: 'DCM'
   (0008, 0104) Code Meaning                         LO: 'Clean Descriptors Op
   ---------
   (0008, 0100) Code Value                           SH: '113107'
   (0008, 0102) Coding Scheme Designator             SH: 'DCM'
   (0008, 0104) Code Meaning                         LO: 'Retain Longitudinal
   ---------
   (0008, 0100) Code Value                           SH: '113108'
   (0008, 0102) Coding Scheme Designator             SH: 'DCM'
   (0008, 0104) Code Meaning                         LO: 'Retain Patient Chara
   ---------
   (0008, 0100) Code Value                           SH: '113109'
   (0008, 0102) Coding Scheme Designator             SH: 'DCM'
   (0008, 0104) Code Meaning                         LO: 'Retain Device Identi
   ---------
   (0008, 0100) Code Value                           SH: '11'
   (0008, 0102) Coding Scheme Designator             SH: 'XNAT'
   (0008, 0103) Coding Scheme Version                SH: '1.0'
```

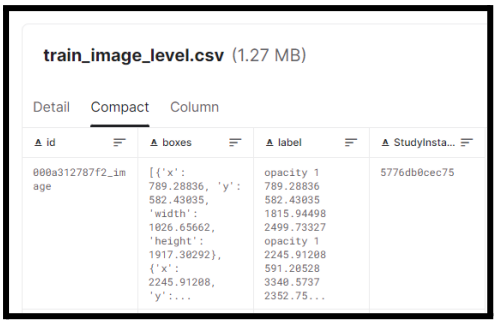
2.2 訓練資料集微觀分析

觀察資料集，發現資料集是多重資料夾的結構，這與資料的標記有關，在train資料夾中為第一層的study_level資料夾，是資料分類的依據，在此資料夾中的所有檔案皆被分類在同一個標記中，而再下一層是存放影像的各個資料夾，是圖片標記的依據，也是我們用於訓練的資料集。



標記資料分為兩個，針對每張影像標記的train_image_level以及針對資料中所分類的study的研究資料標記分類的train_study_level。





在train_study_level中，將資料標記為四種分類：

1. Negative for Pneumonia: 陰性，肺部無混濁的不透明部分。

2. Typical Appearance: 典型COVID-19的外觀，在雙側且周邊瀰漫的混濁且不透明物，伴隨纖維化及肺容積減少。

3. Indeterminate Appearance: 無法判別種類，此類非典型的肺炎，常見特徵是上肺部為主要的混濁(可能為細菌感染、其餘疾病或放射治療)，或只含有單側的混濁。

4. Atypical Appearance: 非典型COVID-19肺炎的外觀，此類並非典

型肺炎及無法判別的種類，此類無肺炎特徵，多為腫塊或結節，或者是疤痕及纖維化。

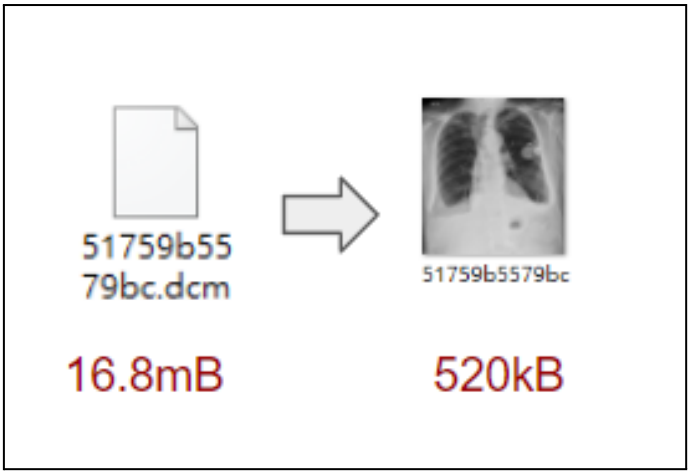
而在tain_image_level中，對影像中肺部標記出不透明部分，用於標記出不正常的區塊。為了對原始資料進一步切分，而我們結合兩個訓練標記資料進行剖析，對於能發現在image中label的標記是對於非陰性的其他三類，針對肺部中的不透明部分框架並標記，多數標記框架的資料為陰性，而標記框架的資料中皆為非陰性的其餘三類。(下圖)。

```
negative_for_NaN is  1736
other_for_NaN is  304
----------------------------
negative_for_label is  0
other_for_label is  4294
```

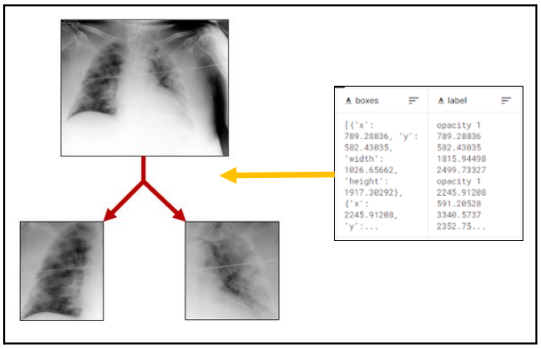
最後得出train_image_level中，boxes及label為非常重要的特徵，不只能夠幫助我們切割出觀察的重點部分，也標記出了在肺部的不透明區塊，是用於分類中重要依據。

2.3 資料前處理

首先，我們著手於如何將.dcm裡的圖片以JPG檔取出，我們發現單獨將圖片取出來使用，而不是使用整個.dcm檔，可以有效的縮小訓練資料的大小(原始大小將近130GB，而取出圖片只占3GB)，我們不斷地測試不同的方法，每次轉換圖片都需要歷經4~5小時將整個資料集的照片取出，將照片取出後檢查檔案的遺失率，以及圖片的完整度。



成功取出照片後依照csv檔的資訊把照片分類成四類，並驗證資料量是否正確、將損毀的資料移除，接著利用資料集中已標記的boxes的標記所提供的起始點位置x,y，及寬度高度w,h對圖片進行切割，切割出肺部部分(下左圖)，在模型訓練時能夠更著重於此部分的特徵進行檢測。也使用許多的方法對訓練集做前處理(下右圖)。




```
transforms = T.Compose([
    #T.Resize((1024, 1024)),
    #T.RandomHorizontalFlip(p=0.5),
    #T.RandomVerticalFlip(0.5),
    #T.RandomRotation(10),
    #T.RandomEqualize(p=1),
    #Cutout(),
    #T.ToTensor(),
    #T.Normalize((0.485, 0.456, 0.406
    #T.GaussianBlur(7,3)

])
```

Chapter 3

實做技術

3.1 模型調整訓練

首先是挑選平台，因為我們自己沒有GPU，所以我們將目光鎖定在了colab與kaggle，實際操作後認為在kaggle上操作kaggle競賽的程式碼方便許多，故選擇在kaggle上執行，但因為GPU使用時間的限制，我們必須在兩個平台上練習，接下來便是選定模型，我們從網路上找到幾組對於影像辨識高評價的模型:1.resNet 2.convnext small 3.convnext large 4.convnext base 5.efficientNet 6. mobileNet，經過爬文對比以及實測後，我們選擇了更輕量化的efficientNet做為我們這次專題的模型。

有三個參數是我們重點調整的對象，分別是:
1.batch size: 透過batch從訓練目標中取樣，來加快ML模型訓練的速度，然而batch size 嚴重受到GPU使用的限制(詳情請見4.4)

2.epochs: 設定epochs總共要用全部的訓練樣本重複跑幾回合，這部分取決於我們有多少時間，但一般狀況下我們無法總是讓筆電長時間的跑，且也有GPU使用時長上的限制，所以觀察epoch該設為多少十分重要 。

3.learning rata: 決定一次走多遠。

有五個function是我們重點使用的對象。分別是:

1.Resize() : 調整大小，至於該調整什麼樣的大小我們有寫一段程式碼，去找出所有圖片的長和高，並加以運用(比如最小的照片是幾乘幾)。

2. RandomVerticalFlip(): 水平翻轉增加資料集

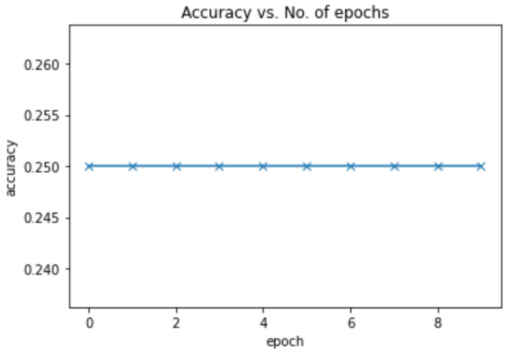
3. Cutout(): 數據增強

4. ToTensor(): 歸一化

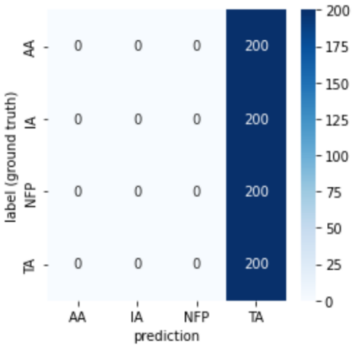
5. Normalize(): 歸一化

3.2 優化模型與整合結果

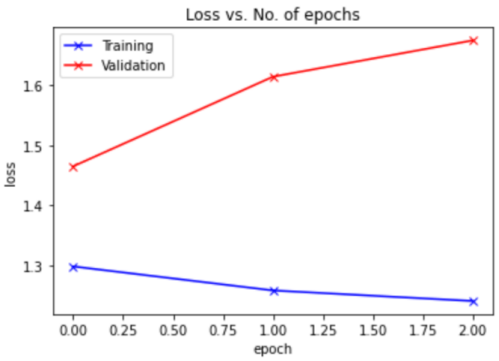
經過前三周的處理後，我們第一次跑出來的第一次跑出來的結果只有25%跟盲猜一樣，然而這很明顯是個失敗的案例，我們從以下三個圖來觀察我們為何失敗，並一次次改進:



(accuracy都維持在0.25，失敗)



(預測都集中在某一類，失敗)



(training loss已擬合，valid loss卻不斷上升，失敗)

經過一次次調整後，我們取得了好上許多的結果。

Chapter 4

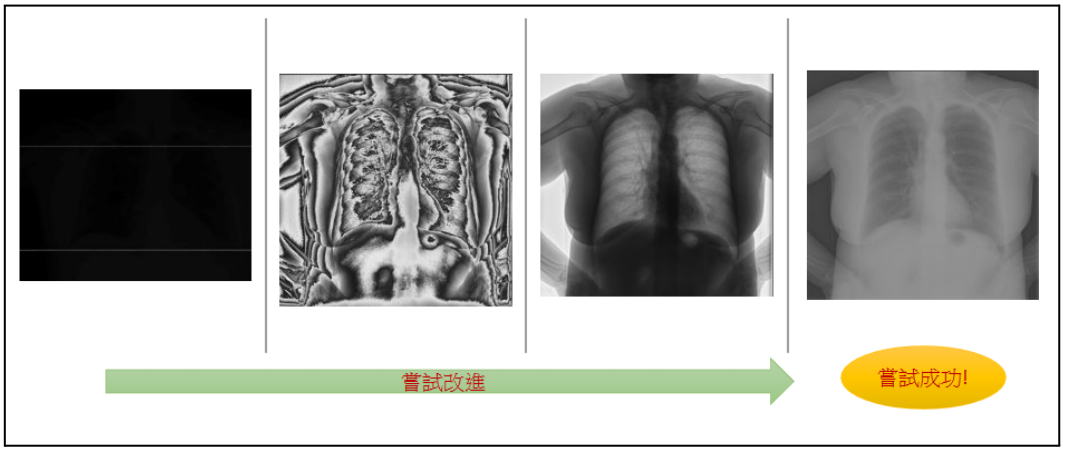
過程中遇到的困難及相應的解決方法

4.1資料集下載失敗

將共128.51GB的資料集下載，經歷了三次失敗，遇到了筆電故障,網路中斷等等的問題，每次嘗試都耗時了7~10小時，最後是靠IDM軟體加上桌機，成功將資料集以更穩定快速的方式下載下來。

4.2 dcm轉jpg檔案轉檔失敗

最剛開始我們透過線上網站將部分.dcm檔裡的圖片轉成.jpg檔，以供觀察，之後我們實做出了可以探索資料集路徑並將.dcm檔全數轉換為.jpg檔的程式，起初取得的照片:雖然能看出一點輪廓但幾乎是全黑的(下圖一)，稍加改進後得到嚴重扭曲圖形(下圖二)，這些很明顯的不是我們要的結果，利用方式調整比如使用.astype()函數，都無法正確地取得，最後調整dtype改為"int16”，得到了較佳的結果，但部分照片也會有灰度相反的問題(下圖三)，我們嘗試了許多種dcm轉jpg的方式，cv2,imageio 等等方法，最終跑出來的結果仍舊是部分照片灰度相反其餘正常，後來利用計算相片中的灰度占比，若黑色占比比較重代表照片出問題(黑色為肺部)，灰度相反的圖片拿出來做灰度顛倒的處理，最後得到正確的結果。



4.3 Kaggle GPU使用時數限制

考量到Kaggle GPU有一周30小時的使用時間限制，我們選擇交叉使用Kaggle與Colab，但不同平台的規則不盡相同，我們花了不少時間熟悉了兩個平台使用上的不同規則後，我們也熟悉了如何在這兩個平台上順練地練習我們的模型。

4.4 Runtime error: cuda out of memory

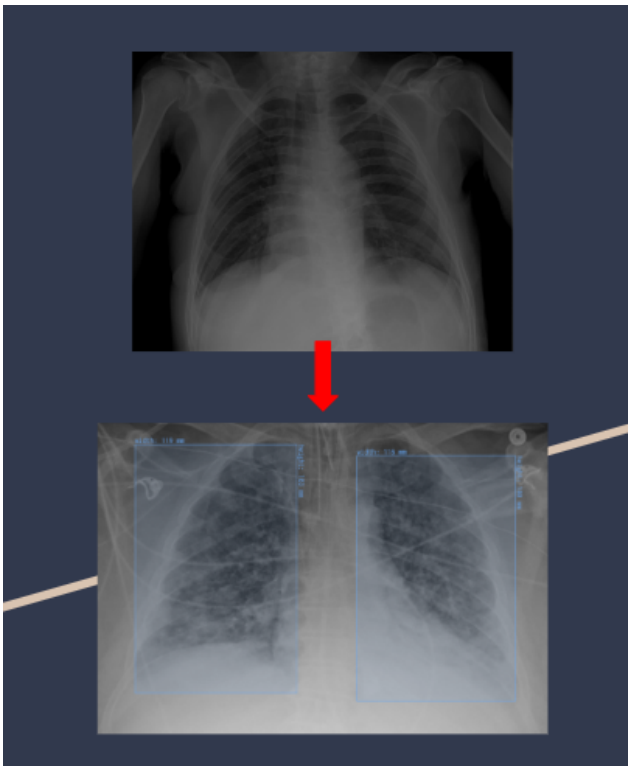
因為GPU影響各個 model 的batch size，而Kaggle的GPU限制使得我們嘗試使用的models普遍只能將batch size設為4，解決方法:選擇輕量化的模型:efficentNet，如此可將batch size 提升到16，雖然16還是不足以跑出最理想的結果，但相比於其他模型，efficentNET是我們找出的最優解。

Chapter 5

最終結果呈現

5.1 圖片切割

有別於剛開始將整張照片進去訓練，我們成功將肺部區塊切割出來，只針對肺部的照片做訓練，下圖為成果:



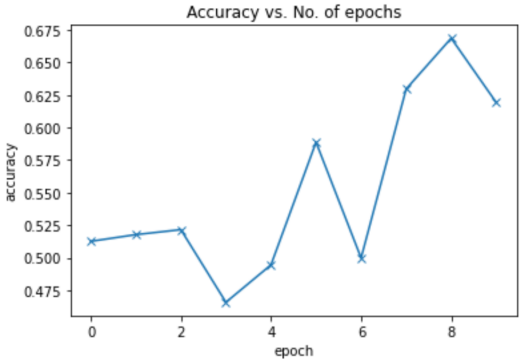
5.2 模型訓練結果

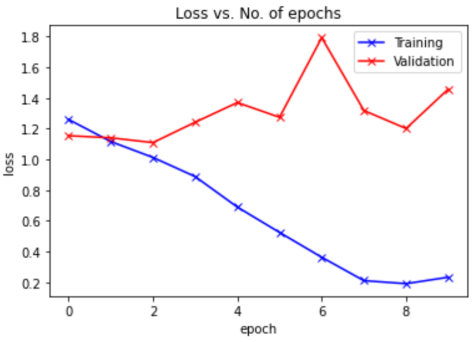
最後得到的結果如下:準確率0.5887(rank:627/1305)。

```
50%|█████     | 5/10 [1:58:50<1:59:12, 1430.53s/it]

['AA', 'IA', 'NFP', 'TA']
[0.36842105 0.26158382 0.67971759 0.84024896]
Plot confusion matrix
Epoch 5: train_loss: 0.5244, val_loss: 1.2744, val_acc: 0.5887
```

在epoch為7,8,9時準確率明顯較高，但透過觀察training loss在epoch7,8,9時持續收斂，有overfitting的現象，我們最終選擇了epoch 5，此外，可以看到valid loss不斷的震盪，可能是因為batch size 較小且類別較多導致它遲遲不收斂，下圖為結果:





Chapter 6

結論與展望

在整個專題的過程中，不管是資料的下載,資料的前處理，還是模型的調整，我們不斷地發現問題並想辦法解決，就算一度得出0.25這樣的模型成果，我們依然不斷嘗試改良，過程相當的費時費力，所幸最終也獲得值得讓我們有成就感的結果，也對如何實做資料科學的模型訓練有了更深的理解。

在未來的目標，會試著了解其他競賽者的研究方式，觀察差異後整合他人的特點並套用在我們自己累積的經驗，也可能會嘗試集成學習（Ensemble learning）使用多種學習算法來獲得比單獨使用任何單獨的學習算法更好的預測性能。