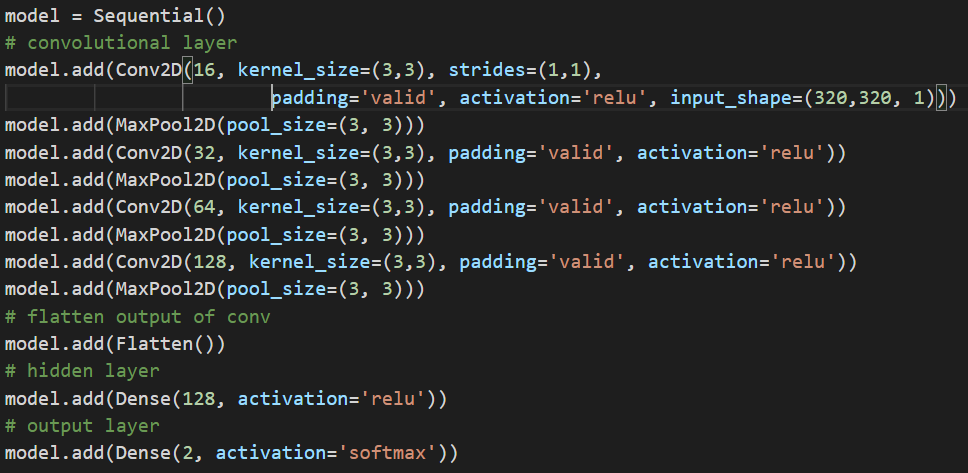
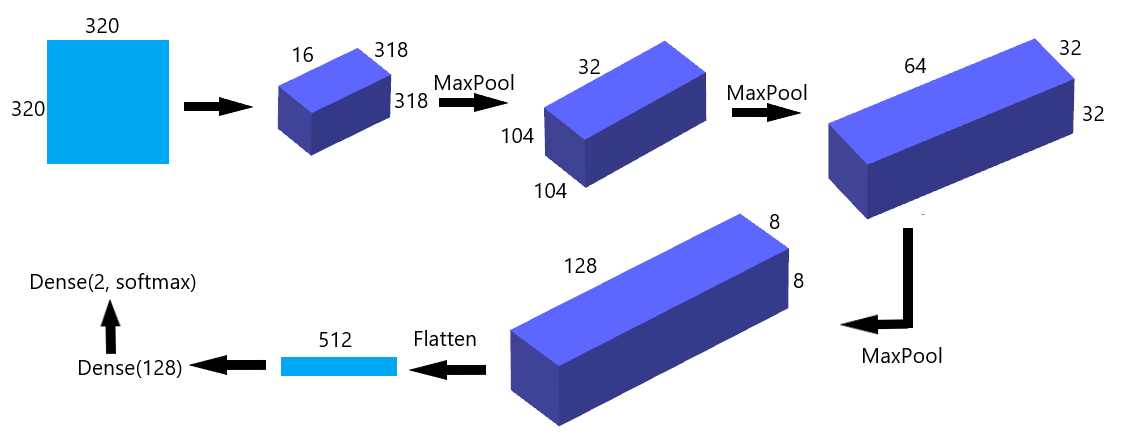
ML\_assignment3

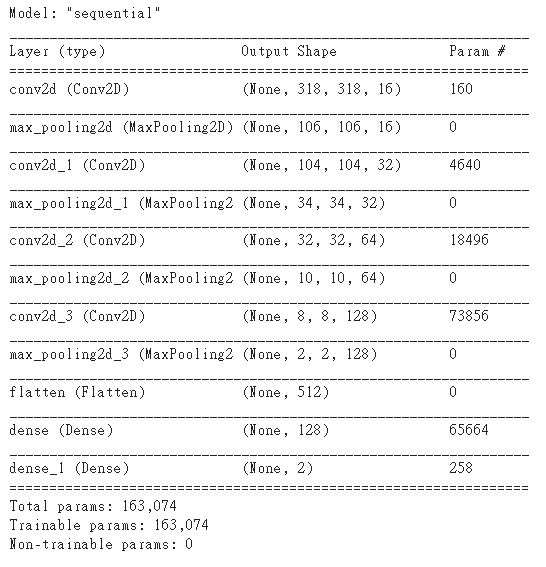
本次作業主要嘗試了兩種方法，一是自己手刻model出來，二是利用pre-trained model，如VCC16、resnet\_v2、Inception2等等。

Method1:手刻model

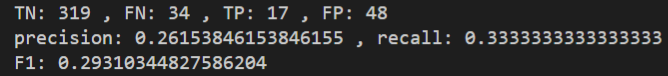




Model Summary:



自己接的效果並不怎麼好，將train data以7:3的比例分為train, test，設定batch\_size = 16、epoch = 20進行訓練，最後得到的F1 score大約為0.3上下。



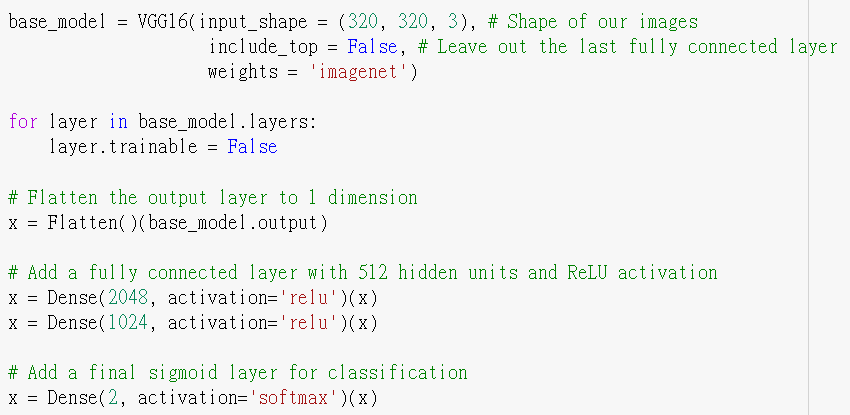
個人認為結果不好的原因大約有幾點：

1. 自己接的Model無法有效地抓取到features。
2. Data的size太小，造成訓練效果不佳

Method2:搭配pre-trained model使用(Transfer-Learning)

搭配pre-trained model使用。利用較有名的pre-trained model，例如：VGG16, ResNet, InceptionV3等，將data丟入pre-trained model後，其輸出再接上自己設定的layer。以VGG16為例：一張含有 文字 的圖片

描述是以非常高的可信度產生



Base\_model為已經pre-trained好的model，因此將其trainable設定成false，單純訓練在pre-trained之後我們所接的Dense(2048)、Dense(1024)、Dense(2)。

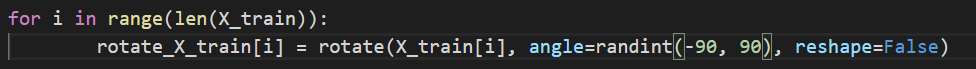
使用pre-trained model時遇到的問題為，大部分的pre-trained model都是以彩色相片進行訓練，與這次作業使用的grayscale image不同，RGB image的shape為(, , 3)，也就是有3個channels，為了解決此問題，在網路上查到的方法是，將grayscale複製成3份，進而當作有3個channels。

Data以與method1一樣的方式進行分割，最後訓練出來的model的performance也大約為0.3。

Data augmentation:

為了讓model更加穩定，並且增加data的數量，嘗試將image進行旋轉、翻轉、縮放等方式，但實際測試下來感覺並沒有什麼效果。

Ex. Image Rotation



複製一組與原data相同的image，將其進行-90度~90度的隨機翻轉，再將翻轉過後的images加入到data中進行訓練。

Bonus

先利用assignment3所做出來的model，以images作為Input data，predict出患者的outcomes，然後將outcomes當作HER中的其中一項attribute，也就是將CNN model的輸出當作40幾樣attributes中的其中一樣，再用這些attributes來對當初使用的Random Forest做訓練，以得到更好的結果。

當初在做assignment2時，self-test的F1大約都為0.5多一點，最後實際的F1為0.51，算是self-test跟實際test的結果蠻相當的。

利用CXR搭配EHR進行訓練，在self-test時大約可以達到0.7的F1 score。比起單純EHR多了快0.2。

