

# 影像處理期末作業

資訊所碩一 P76081116 鄭皓中

## Questions

這次的問題是要進行 **vertebra segmentation**，以用來後續進行相關應用，像是偵測 **scoliosis** 等問題。

主要問題有兩個：

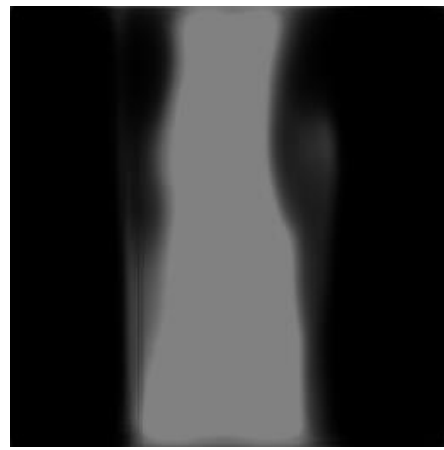
1. 如何在不多的影像(60 張 **source**、60 張 **label**)及每個影像的條件都不一樣(明暗、對比等)的情況下對 **vertebra** 進行 **segmentation**
2. 如何將每一個 **vertebra** 分別做 **DC evaluation** 及 **DC of average**

## Methods

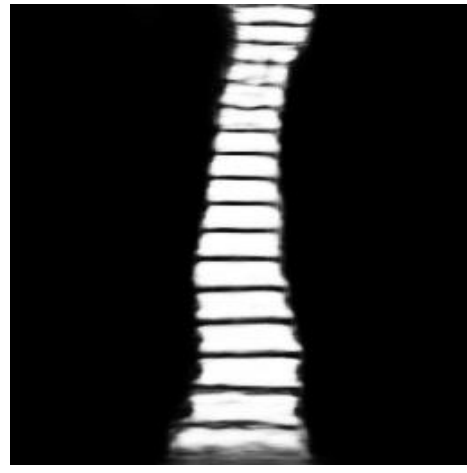
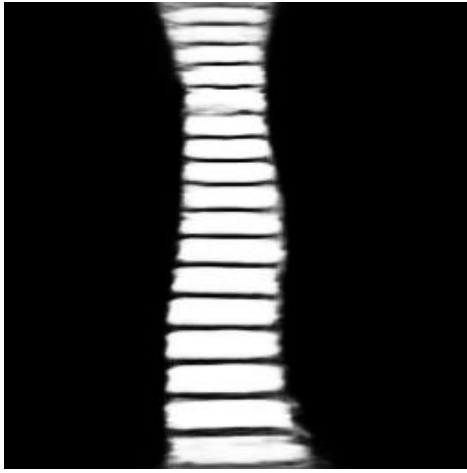
1. 如何在不多的影像(60 張 **source**、60 張 **label**)及每個影像的條件都不一樣(明暗、對比等)的情況下對 **vertebra** 進行 **segmentation**

這次選擇採用 **deep learning** 的方法進行切割，採用的是基本應用 **encoder-decoder** 架構的 **U-Net**，會選用此架構的原因在於 **U-Net** 非常適合用於醫學影像的切割，因為醫學影像本身具有資料量較少且醫學影像型態較統一等特點，善用這兩點的 **U-Net** 切割可以達到很好的效果。

然後選定方法之後遇到了許多的問題，一開始 **train** 出來的 **model**，用來預測的效果都很差，如下面兩張圖：



這兩張圖的外型其實跟原圖蠻相近的，就是裡面的細節都不見了。後來持續思考改良，發現原因可能在於前面提到的影像品質不一致有關，所以我開始把 60 張 **source** 的影像做前處理，有用到 **Gamma Correction** 及 **Histogram Equalization** 等，後來發現對每張影像做 **Histogram Equalization** 後發現效果改善了很多，如下面兩張圖：



細節已經完全顯示出來了，但是預測出來的比例與原圖不同，之後只要對預測出來的結果做 **resize** 就可以保持原圖的比例了。

順便附帶 **source** 做 Histogram Equalization 前後的結果：(左圖是前，右圖是後)



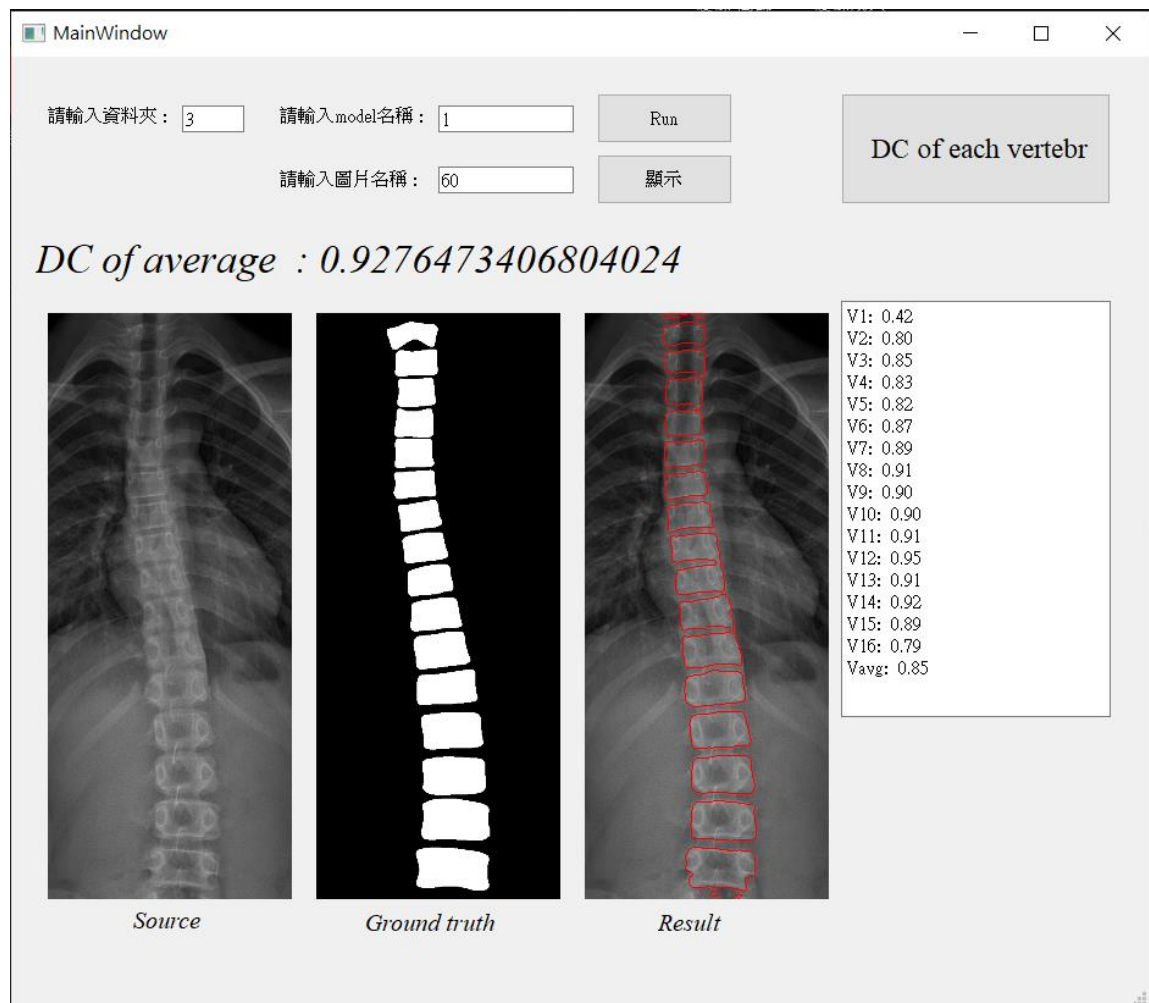
## 2. 如何將每一個 **vertebra** 分別做 DC evaluation 及 DC of average

由於預測出來的圖是屬於一整體的，若需要將每一節的 **vertebra** 進行評估，需要將每一節的位置在哪裡都找出來，上網找了許多的方法，最後找出了利用對 **ground truth** 找到 **connected component** 的每個部件類別座標位置後，在依照同一個類別位置進行評估，這樣可以有效考慮到每一截 **vertebra**

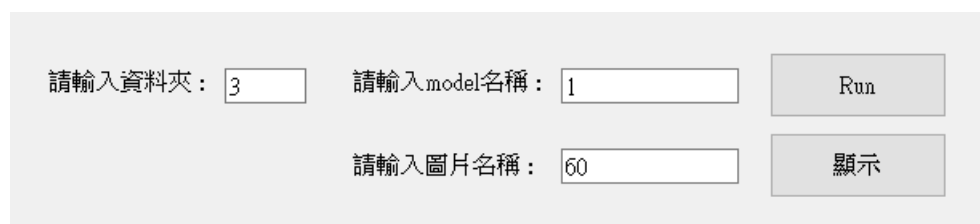
```
V1: 0.42
V2: 0.80
V3: 0.85
V4: 0.83
V5: 0.82
V6: 0.87
V7: 0.89
V8: 0.91
V9: 0.90
V10: 0.90
V11: 0.91
V12: 0.95
V13: 0.91
V14: 0.92
V15: 0.89
V16: 0.79
Vavg: 0.85
```

# Results

## GUI 介面

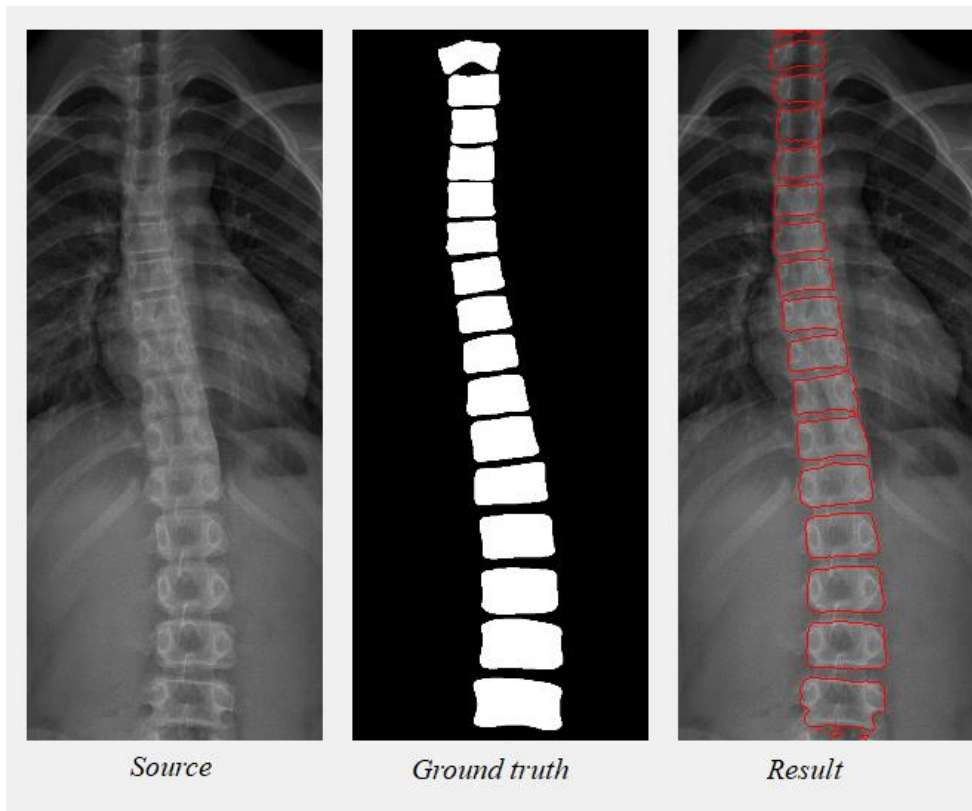


此介面為本次 final project GUI



上面的部分有三個可以輸入數字的地方，分別是輸入資料夾名稱、model 名稱、圖片名稱，還有兩個按鈕：run 跟 顯示。

首先輸入完資料夾名稱、model 名稱後，請先按 run 跑預測，之後再根據輸入的資料夾名稱，輸入圖片名稱的編號，才能按顯示，顯示結果的圖片。



顯示結果的圖片如上圖所示，介面的左圖是原圖，即要分析的影像，中間為此影像相對的 **ground truth**，右圖為此原圖分析過後的結果，紅色線部分為預測的 **mask**。

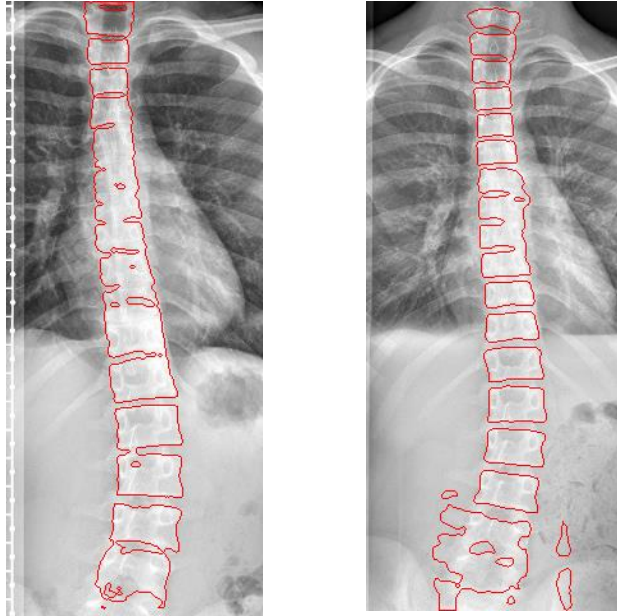
接著最右邊有一個按鈕 **DC of each vertebra**，按下即會輸出每一節 **vertebra** 的切割結果，是採用 **DC evaluation** 的方式評估，每一節分別計算後會輸出一個總體平均的分割值。(而介面的 **DC of average** 是整張圖去評測的結果)

**DC of each vertebra**

V1: 0.42  
V2: 0.80  
V3: 0.85  
V4: 0.83  
V5: 0.82  
V6: 0.87  
V7: 0.89  
V8: 0.91  
V9: 0.90  
V10: 0.90  
V11: 0.91  
V12: 0.95  
V13: 0.91  
V14: 0.92  
V15: 0.89  
V16: 0.79  
Vavg: 0.85

## Discussion

發現有些部分的預測出來的紅線沒有每截 **vertebra** 都分得很清楚，如下圖所示：



我覺得可能是跟我把每張預測完的圖做二值化處理後，利用邊界畫出紅線，而二值化處理所用的閾值可能無法普遍套用到每一張預測的結果上，所以才會有紅線連在一起的問題(但是這只是顯示的問題，與 **dc** 的準確度應該無關)。而圖形下半部預測比較不準確的原因可能是 **model** 學得較不佳，因為 **model** 學習時是學習特徵，原圖的影像的底部背景為白色而上半部背景為黑色，因此上半部特徵較為強烈，下半部較模糊，所以 **model** 學得比較不好。

## Conclusion

這次期末作業讓我體會到對影像做前處理很重要，也讓我體悟到試誤學習的重要性，如果沒有一開始直接拿原圖去 **train**，不知道 **train** 出來的結果會這麼不好，如此才能再次思考資料的特性，然後去做相對應的前處理。還有學習到如何尋求問題的解決方案，上網查詢方法以及與同學的相互學習討論，教學相長的过程真的很不錯，這次的期末作業真的是獲益良多。