機器學習 專案作業二

組員

M11221004 侯郡凌

M11223040 邱琳恩

中華民國一一三年五月二日

摘要

本研究旨在探討氣管圖像分割領域中使用不同深度學習模型的應用。動機在於提高醫療影像分析的準確度，尤其是在氣管內管圖像分割方面，以改善臨床診斷和治療決策。為達到此目的，本研究採用Unet和Unet++兩種模型進行實驗，使用不同的超參數組合，以評估其在氣管內管圖像分割方面的績效，且為了解決過擬合問題，本研究採用L2正則化、Dropout等技術。實驗結果顯示Unet和Unet++模型整體表現不如預期，可能受到資料集規模、模型架構、資料處理方式和超參數設置等因素的影響，為提升模型性能，需要進一步的嘗試和改進。未來的改進方向包括利用套裝程式自動尋找最佳超參數組合和考慮使用預訓練模型作為基礎架構，以提高模型的準確性和泛化能力。

關鍵字：機器學習、圖像分割、Unet、CNN、分割氣管內管圖像

一、 緒論

1.1 動機

本研究旨在了解不同的圖像分割模型在ETT\_v3資料集上的實際表現，透過CNN圖像分割技術可針對影像進行準確且快速的分析，而在本研究中透過該方式可以協助醫生定位氣管位置，以提升病患的醫療體驗。近年來，CNN圖像分割技術在醫學影像處理領域展現出顯著的效果，以下是透過深度學習進行氣管內管圖像分割的主要動機：

1. 高準確性：透過影像分割模型能夠精確抓取影像的細節，從而提高分割精確度，這對於氣管內管的正確辨識和定位極為重要。
2. 適用多種影像對比度：深度學習模型能夠有效處理不同對比度的影像，實現穩定的分割表現。
3. 有效使用有限標註數據：模型能夠從有限的標註數據中學習到豐富的特徵，這對於醫學影像分割領域相當重要。
4. 促進臨床應用：準確的氣管內管分割有助於醫生進行更加準確的診斷和治療決策，可進一步促進自動化診斷系統的開發，提高臨床工作效率和病人安全。

總結，選擇作為氣管內管圖像分割的深度學習架構，基於它們在精確度以及適應性方面的優異表現，以及對有限標註數據高效率的學習能力，透過這些技術的應用，期望能夠促進醫生做出更準確的醫療決策。

1.2 目的

本研究旨在利用ETT\_v3資料集建立氣管內管圖像分割模型，採用卷積神經網路（CNN）進行模型訓練，其主要目的如下：

1. 提升分割精確度：利用先進的深度學習模型，實現高準確性的氣管內管分割，確保醫療影像分析的可靠性。
2. 提供臨床決策：精確的氣管內管分割結果可為醫師提供重要參考，在需要快速且準確判斷的緊急情況下，協助他們作出正確診斷和醫療決策。
3. 降低人為誤差：自動化分割可以減少實際操作過程中可能出現的誤差，降低因人為判斷錯誤對患者造成的風險。
4. 提高醫療效率：自動化技術可以節省醫生分析影像的時間，提高工作效率，使他們能夠專注於患者的其他治療需求。

二、 方法

本研究採用Unet以及Unet++作為氣管圖像分割模型，旨在透過深度學習技術實現高度準確的影像分割，程式架構主要包含資料集的讀取、影像與遮罩的前置處理、模型建置、模型訓練、以及模型績效評估等步驟。

**針對Unet模型:**首先，透過os模組定義資料夾路徑，載入資料集內容，將圖像轉為適當的256\*256大小和灰階格式。並對於訓練集圖像進行影像增廣技術，將原影像擴增至兩倍，使用了隨機翻轉、對比度增強、旋轉和平移等增強方法。接著，選用了Unet的深度學習架構來執行圖像分割任務，模型建立過程中包含了卷積、池化、Dropout等操作，並使用binary cross entropy作為損失函數進行訓練。而在模型績效評估部分選用IoU（Intersection over Union）作為指標，用以衡量模型對於圖像分割任務的準確度。最後輸出原始影像、真實遮罩以及預測遮罩，以便進一步分析。

**針對Unet++模型:**首先，載入影像與遮罩並統一調整大小為256\*256像素，以確保統一的訓練格式。對於訓練集影像，採用影像擴增技術，將原始影像擴增至兩倍，擴增方法包括隨機旋轉、對比度增強、縮放、亮度調整等，以擴充訓練數據並提升模型的泛化能力。接著，建構Unet++模型，並應用正則化技術，如L2正則化、early stopping以及Dropout技術，以防止模型過度擬合訓練數據，確保模型的泛化性能。在模型訓練過程中，採用IoU（Intersection over Union）作為主要的績效評估指標，以評估模型的分割準確度。最後，模型的輸出包括原始影像、真實遮罩以及預測遮罩，以便進一步分析。

三、 實驗

3.1 資料集

本研究使用ETT\_v3資料集為氣管內管影像資料集，該資料集包含了5個子資料集，分別為Fold1 Fold2、Fold3、Fold4、Fold5，且每個Fold分別包含了train、trainannot、test、testannot、val、valannot子資料集，train、test、val皆為jpg格式，trainannot、testannot、valannot皆為png格式。

以Fold1為例，train資料集資料筆數為287，trainannot資料集資料筆數為287，test資料集資料筆數為47，testannot資料集資料筆數為47，val資料集資料筆數為47，valannot資料集資料筆數為47，如表1所示。

表1、 ETT\_v3資料集 (以Fold1為例)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 資料集名稱 | 資料定義 | 資料型態 | 數量 | 資料1 | 資料2 |
| train | 訓練集 | jpg | 287 |  |  |
| val | 驗證集 | jpg | 47 |
| test | 測試集 | jpg | 47 |
| trainannot | 訓練目標遮罩 | png | 287 |  |  |
| valannot | 驗證目標遮罩 | png | 47 |
| testannot | 測試目標遮罩 | png | 47 |

3.2 前置處理

本研究針對兩種不同的分割模型進行了不同的資料前置處理，分別針對Unet與Unet++進行了細緻的處理。

**針對Unet模型:**首先，使用了resize()方法將圖像調整為256\*256大小，並將圖像轉換為灰階模式。且為了增加訓練集的圖片數量，針對train資料集使用圖像增強技術，包含隨機旋轉（以中心旋轉-30至30度）、對比度增強（隨機增強0.8-1.2倍）、隨機水平和垂直位移等強化方式，最後，將訓練數據擴增至原影像的一倍。在此過程中，其餘資料集數量保持不變。

**針對Unet++模型:**首先，使用cv2套件的resize()方法將影像以及遮罩的大小統一調整為256\*256，以確保統一的訓練格式。然後，對訓練集(train)影像進行了影像擴增處理，使原始影像擴增至兩倍，其中包括隨機旋轉（以中心旋轉-30至30度）、對比度增強（隨機增強1.0-1.5倍）、縮放大小（將影像縮放至0.8-1.2倍）、亮度調整（隨機選擇亮度0.5-1.5）等強化方式，使用random.choice() 隨機選取一種強化方式進行擴增，最後，將訓練數據擴增至574張。在此過程中，其餘資料集數量保持不變。接著，在輸入模型前，對影像進行正規化處理，即所有像素值除以255，將像素值縮放至0-1的範圍內；而對於遮罩，像素值大於0的設置為1，而像素值小於0的設置為0，以確保像素值範圍在0至1之間。最後，將影像與遮罩轉換為一維向量格式輸入至模型中進行訓練。

3.3 實驗設計

在本研究中使用訓練集進行模型的訓練，同時使用驗證集在訓練過程中進行驗證，在訓練過程中，保存驗證集中表現最佳的權重，並應用early stopping技術提早停止以及Dropout技術隨機丟棄一些神經元，並且繪製訓練集與驗證集的損失函數曲線，以監控模型的收斂情況和穩定性。

最後，本研究使用最佳的權重進行測試評估，並以IoU作為主要的評估指標，通過調整超參數組合，如batch size、epoch、優化器、損失函數、L2的λ值以及Dropout rate等進行一系列實驗，以找出每個子資料集的最佳的超參數組合，以最大程度優化模型。

**針對Unet模型:**本研究以Fold1進行實驗，選出最合適的超參數配置，如表2所示，可發現在該任務中整體的IoU分數偏低，且其中較大的Epochs和Batch\_size沒有帶來更好的績效表現，因此本研究在Unet圖像分割任務中選用較小的Epochs和Batch\_size組合，而在L2正則化技術部分，可發現對於較小的參數組合來說，L2技術無法提升績效表現，因此最終選用第一組超參數配置進行其餘Fold的實驗。

**針對Unet++模型:**首先使用同一子資料集即Fold3進行實驗，根據表3的結果顯示，當Batch size設置為4、Epoch設置為20、優化器選擇Adam、損失函數為Dice損失函數、Dropout rate設置為0.4，以及L2正則化的參數（λ值）設置為0.001時，模型的績效（IoU）表現最佳。雖然測試績效僅達到9.3％，本研究認為可能是由於資料量不足以及影像缺乏多樣性所導致。值得注意的是，當Batch size設置為2時，模型的績效約為8.7％，這表示目前模型的Batch size可以設置為2或4，並且本研究還嘗試將損失函數更換為二元交叉熵，根據實驗結果，績效並未有所提升反而下降。因此，本研究決定繼續使用Dice損失函數於不同子資料集實驗。

表2、Unet：同一子資料集分割實驗結果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fold | Batch  size | Epochs | Optimizer | Loss function | Dropout  rate | L2 | Train  Mean IoU | Test  Mean IoU |
| 1 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | x | **0.011971** | **0.009274** |
| 1 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | 0.1 | 0.011878 | 0.009202 |
| 1 | 4 | 5 | Adam | binary\_cross  entropy | 0.5 | 0.1 | 0.011911 | 0.009192 |
| 1 | 8 | 10 | Adam | binary\_cross  entropy | 0.5 | 0.1 | 0.011867 | 0.009160 |
| 1 | 8 | 10 | Adam | binary\_cross  entropy | 0.5 | x | 0.011799 | 0.009069 |
| 1 | 16 | 15 | Adam | binary\_cross  entropy | 0.5 | x | 0.011709 | 0.008931 |

表3、Unet++：同一子資料集分割實驗結果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fold | Batch  size | Epochs | Optimizer | Loss function | Dropout  rate | L2 | Train  Mean IoU | Test  Mean IoU |
| 3 | 4 | 20 | Adam | Dice Loss | 0.2 | 0.001 | 0.0745 | 0.0820 |
| 3 | 4 | 20 | Adam | Binary\_crossentropy | 0.2 | 0.001 | 0.0241 | 0.0136 |
| 3 | 4 | 20 | Adam | Dice Loss | 0.3 | 0.001 | 0.0739 | 0.0370 |
| 3 | 2 | 20 | Adam | Dice Loss | 0.4 | 0.001 | 0.0745 | 0.0871 |
| 3 | 4 | 20 | Adam | Dice Loss | 0.4 | 0.001 | **0.0750** | **0.0938** |
| 3 | 4 | 20 | Adam | Binary\_crossentropy | 0.4 | 0.001 | 0.0217 | 0.0044 |

3.4 實驗結果

**針對Unet模型:**如圖1所示，可以發現該模型無法準確地進行圖像的預測，在Predicted Mask的圖像中，只有在左下角可發現有跑出微小的預測白色線條，與Ground Truth Mask圖像不符，且如同表4所示，各個Fold的整體績效也偏低，且在實驗中發現，Fold3使用L2技術時可提升些許績效表現，與先前針對Fold1的預測結果是相反的，且在分割平均績效部分，可以發現在Unet模型中，整體的績效表現都不佳，綜合上述結果，本研究推斷了幾種導致Unet模型績效不佳的可能性:

1. 資料處理:在進行圖像資料前處理時，可能使用不恰當的處理方式導致訓練數據品質下降，進而影響模型訓練的成果。
2. 訓練數據數量:在進行圖像增強時，可能因為產生的圖片數量不足，讓模型無法有效地進行訓練，導致學習績效不佳。
3. 模型架構與超參數:在模型建置部分，可能需要更複雜的網絡來捕捉圖像特徵，此外，在超參數設定部分，需要嘗試更多超參數組合，以找出最適當的配置進行模型訓練。
4. 模型訓練時間:在模型訓練時間部分，因訓練時間對於模型性能有極大影響，可能需要增加更多訓練時間，讓模型充分訓練，以提高模型的性能。

綜合結果來說，本研究認為這些因素都有可能導致Unet在圖像分割任務中績效表現不佳，需要進行進一步的調整和優化，以達到最佳的模型績效。

**針對Unet++模型:**以Fold3的最佳超參數組合作為基準，在不同子資料集上進行了不同參數的實驗，如表5所示，不同子資料集的超參數組合不盡相同，使用Fold3的最佳超參數組合在其他子資料集上的表現不一定為最佳，因此進一步調整了一些超參數，如Batch size、Dropout rate (20%-40%)以及L2的參數（λ值）。其中，可以發現子資料集Fold1的績效在此模型表現最佳，測試的IoU績效達到13%，五個子資料集的平均訓練以及測試績效分別為9.23%以及9.43%。

最後，輸出模型分割圖，如圖2所示，從左至右依序為：增強後的影像、真實遮罩、預測遮罩。

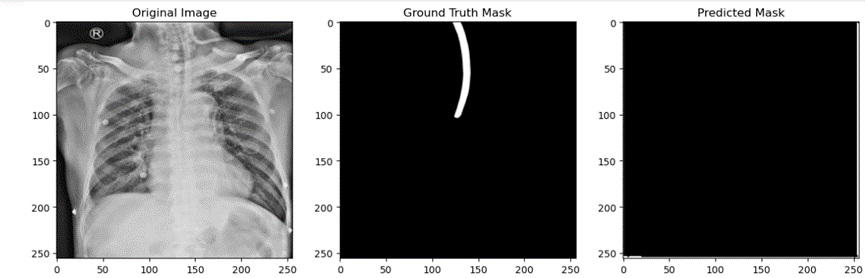


圖1、Unet分割實驗結果

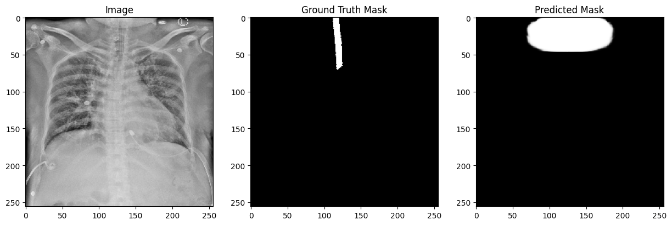


圖2、Unet++分割實驗結果

表4、Unet：不同子資料集分割實驗結果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fold | Batch  size | Epochs | Optimizer | Loss function | Dropout  rate | L2 | Train  Mean IoU | Test  Mean IoU |
| 1 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | x | 0.011971 | 0.009274 |
| 2 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | x | 0.011922 | 0.009999 |
| 3 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | 0.1 | 0.012005 | 0.009673 |
| 4 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | x | **0.012038** | **0.010002** |
| 5 | 2 | 5 | Adam | binary\_cross  entropy | 0.5 | x | 0.012304 | 0.009901 |
| 平均IoU |  |  |  |  |  |  | 0.012048 | 0.009770 |

表5、Unet++：不同子資料集分割實驗結果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fold | Batch  size | Epochs | Optimizer | Loss function | Dropout  rate | L2 | Train  Mean IoU | Test  Mean IoU |
| 1 | 4 | 20 | Adam | Dice Loss | 0.2 | 0.0001 | **0.1206** | **0.1307** |
| 2 | 2 | 20 | Adam | Dice Loss | 0.2 | 0.001 | 0.0759 | 0.0721 |
| 3 | 4 | 20 | Adam | Dice Loss | 0.4 | 0.0001 | 0.1087 | 0.1132 |
| 4 | 4 | 20 | Adam | Dice Loss | 0.3 | 0.001 | 0.0767 | 0.0766 |
| 5 | 2 | 20 | Adam | Dice Loss | 0.2 | 0.0001 | 0.0795 | 0.0791 |
| 平均IoU |  |  |  |  |  |  | 0.0923 | 0.0943 |

四、 結論

本研究針對氣管分割資料集運用不同分割模型進行了深入的實驗。對於Unet模型，可發現整體績效不如預期，也從整體的結果中推斷了幾種導致績效不如預期的可能性，如不恰當的處理方式導致數據品質下降、訓練數據數量不足、模型架構與超參數配置不佳、模型訓練時間不夠，綜合上述，本研究認為針對Unet模型部分，仍需進一步的調整和優化，才能達到最佳的模型績效。然而，對於Unet++模型，本研究發現整體績效較低，這可能為不同因素所導致。首先，資料集的規模相對較小，可能限制模型學習到的特徵有限，因而無法進行準確的分割。其次，模型本身的架構可能存在不足，例如層數過少或建構錯誤，未能準確切割複雜的氣管結構。此外，對資料集進行處理的方式和超參數的設置也可能影響了模型的績效。

未來本研究計劃採取一些措施來改善模型的績效。首先，會考慮使用套裝程式來幫助模型自動尋找最佳的超參數組合，這樣可以節省大量的實驗時間，並且提升模型的績效。其次，將使用現有的預訓練模型作為基礎架構，可以讓模型使用預訓練的權重，從而提高模型的準確性和泛化能力，同時也可以避免模型構建不足的問題。本研究也認為，除了採用IoU作為績效指標外，在未來的實驗中也可納入平均誤差公分、誤差在0.5公分內準確率、誤差在1.0公分內準確率用於評估模型性能，有助於擴展對模型績效的評估，使評估更加全面。

參考文獻

|  |
| --- |
| GeeksforGeeks. (2023, July 31). Unet Architecture explained. Geeksfor-Geeks. https://www.geeksforgeeks.org/unet-architecture-explained/ |
| I code so i am. (2020, September 20). Day 20：使用 U-Net 作影像分割(Image Segmentation). IT邦幫忙. https://ithelp.ithome.com.tw/articles/10240314 |
| Liu, C. H. (2023, May 2). *深度學習Paper系列(05)：U-Net*. Medium. https://tomohiroliu22.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92paper%E7%B3%BB%E5%88%97-05-u-net-41be7533c934 |
| Margaret Maynard-Reid. (2022, February 19). *U-Net Image Segmentatio-n in Keras*. PyImageSearch. https://www.machinelearningnuggets.com/image-segmentation-with-u-net-define-u-net-model-from-scratch-in-keras-and-tensorflow/ |
| 李謦伊. (2021, May 27). 影像分割 Image Segmentation — 語義分割 Semantic Segmentation(1). Medium. https://medium.com/ching-i/%E5%BD%B1%E5%83%8F%E5%88%86%E5%89%B2-image-segmentation-%E8%AA%9E%E7%BE%A9%E5%88%86%E5%89%B2-semantic-segmentation-1-53a1dde9ed92 |
|  |