國立成功大學

資訊工程學系

影像處理

期末專題報告

***Segmentation of Carpal Tunnel***

***from magnetic resonance image (MRI)***

學生：P76094169 黃仁鴻

授課老師： 孫永年 教授

中華民國110年1月

# 目錄

[目錄 i](#_Toc61207039)

[表目錄 ii](#_Toc61207040)

[圖目錄 ii](#_Toc61207041)

[一、問題描述 1](#_Toc61207042)

[三、研究方法 2](#_Toc61207043)

[3.1 資料前處理 2](#_Toc61207044)

[3.2 Loss Function 4](#_Toc61207045)

[四、模型架構 5](#_Toc61207046)

[五、成果與討論 7](#_Toc61207047)

[5.1開發環境 7](#_Toc61207048)

[5.2實驗討論 7](#_Toc61207049)

[參考文獻 8](#_Toc61207050)

# 表目錄

[表 1深度學習開發環境 7](#_Toc61206493)

[表 2 GUI開發環境 7](#_Toc61206494)

[表 3 實驗結果 7](#_Toc61206495)

# 圖目錄

[圖 1 step 1，切下中央 384 x 384 的範圍 2](#_Toc61206501)

[圖 2 step 2，裁切出9個位置的影像 3](#_Toc61206502)

[圖 3 step 3，將步驟2所得到的影像旋轉四種角度 3](#_Toc61206503)

[圖 4 VoVNet 5](#_Toc61206504)

[圖 5 實驗模型 6](#_Toc61206505)

# 一、問題描述

本次研究透過深度學習的方法，使用核磁共振成像所取得之T1與T2兩張影像作為輸入，來分割出 腕隧道(Carpal tunnel, CT)、屈肌腱(Flexor tendons, FT)以及 正中神經(Median nerve, MN) 三個類別。

在研究過程中遇到的以下兩點問題將會在研究方法中說明如何處理：

1. 資料量過少
2. 正中神經無法辨識

# 三、研究方法

### 3.1 資料前處理

本次實驗可使用的資料為207份大小為512 x 512 pixel的 T1、T2、CT、FT與MN影像，但使用深度學習方法時面臨到資料量過少的問題，且一張影像上許多地方都是為拍攝到物體的黑色區域，因此先對資料進行前處理與資料擴充，處理流程如下：

1. 在前處理時會先將四周裁切掉64 pixel的寬度(如圖1)
2. 以256 x 256 pixel的大小切出 左上、上、右上，左、中、右 與 左下、下、右下這九個位置(如圖2)
3. 將切出的圖片分別旋轉 、、、 (如圖3)

經過這三個步驟後就可將資料量擴增至36倍。



圖 1 step 1，切下中央 384 x 384 的範圍

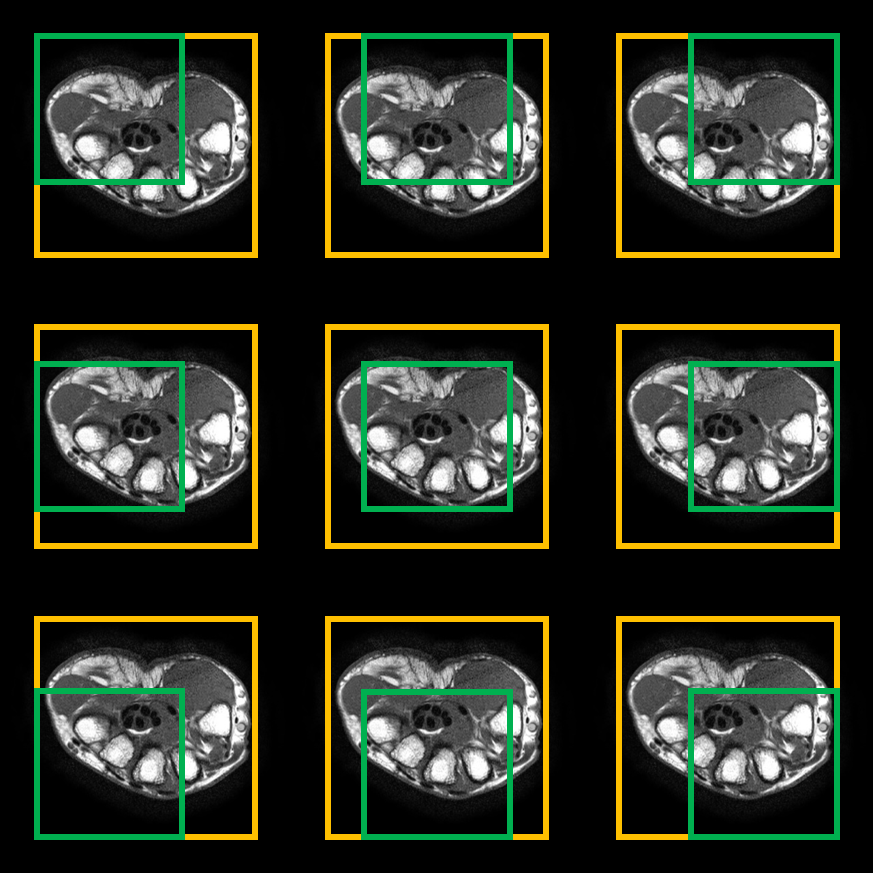


圖 2 step 2，裁切出9個位置的影像

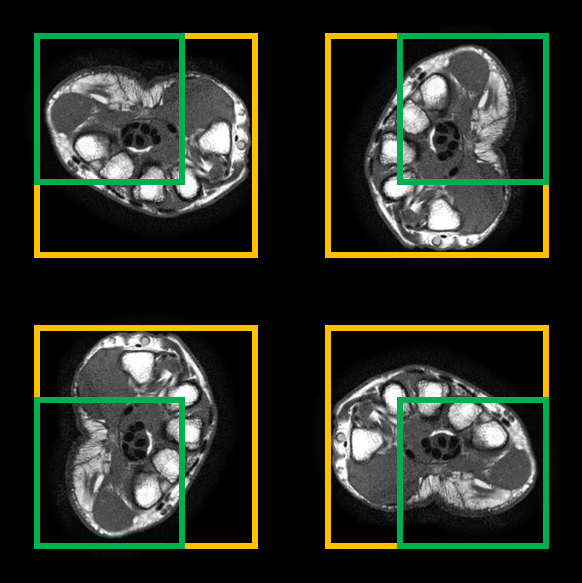


圖 3 step 3，將步驟2所得到的影像旋轉四種角度

### 3.2 Loss Function

在一開始的實驗中使用的損失函數是Binary Cross Entropy(BCE) Loss來對每個pixel進行分類，但這個loss卻無法偵測到MN，原因在於MN只占整張影像面積的一小部分，就算沒能成功分類也不會有太大的影響。為了處理這個問題，實驗將BCE Loss改成Dice Coefficient(DC) Loss與自訂的Weighted BCE Loss。

#### 3.2.1 Dice Coefficient Loss

式一為DC Loss的計算公式：

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

如果是使用DC Loss便可以避免掉MN偵測不到的問題，但卻會使得切割形狀破碎不連貫。Weighted BCE Loss便是為了改善這個問題而加入的

#### 3.2.2 Weighted BCE Loss

式二為一般BCE Loss的計算公式：

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

而Weighted BCE Loss則改成式三的公式

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

其中的w表示label切割面積占整張圖片的比例，如果占比較低，便會得到較大的權重。利用這種方式來強化像是MN這種小區塊的重要性，但缺點是會切割出過大的面積。因此使用DC Loss佔0.7與Weighted BCE Loss佔0.3的比例混和Loss進行訓練。

# 四、模型架構

在最開始的實驗是使用U-Net [1]作為模型架構，但並沒有獲得良好的成果。因此後續將U-Net改成由FCN [2]與VoVNet [3](如圖4)組成的模型。

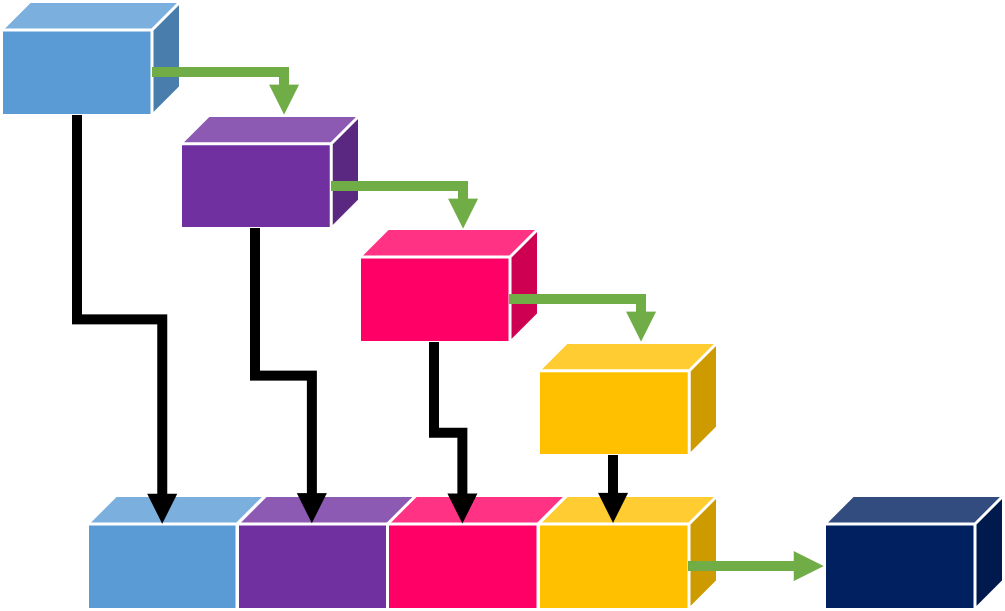


圖 4 VoVNet

圖5是本次實驗使用的模型架構圖，在最開始由Encoder獲得多尺度的特徵，並將各尺度的特徵通過pixel shuffle(sub-pixel [4])成128 x 128後串接在一起後進入VoVNet的區塊，最後再通過一次pixel shuffle成256 x 256並加上最開始的後進行分類。

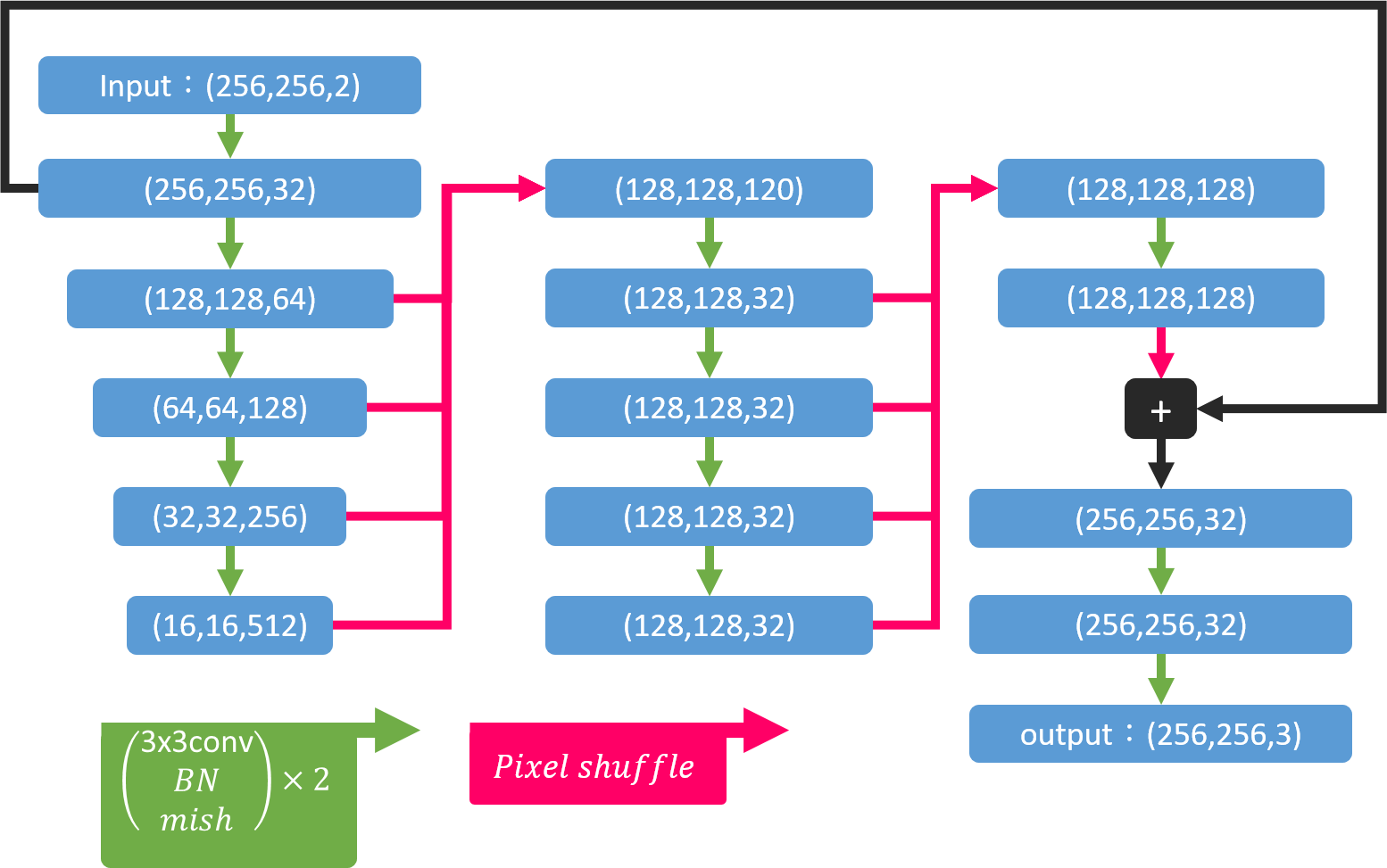


圖 5 實驗模型

# 五、成果與討論

### 5.1開發環境

表 1深度學習開發環境

|  |  |
| --- | --- |
| **深度學習環境** | Google Colab |
| **使用語言** | Python3 |
| **深度學習函式庫** | pytorch 1.7.0+cu101 |

表 2 GUI開發環境

|  |  |
| --- | --- |
| **作業系統** | Windows 10 專業版 Insider Preview，OS組建：21286.1000 |
| **編輯器** | VSCode + rust-analyzer |
| **程式語言** | Rust |
| **編譯器** | Rustc 1.49.0 |
| **GUI函式庫** | iced 0.2 |
| **深度學習函式庫** | tch-rs 0.3.1，搭配libtorch 1.7.1 cpu |

### 5.2實驗討論

訓練時所使用的optimizer是Adamax(使用預設的參數設定)，loss則是0.3 Weighted BCE Loss + 0.7 DC Loss，每次訓練都會經過5個epoch。最後驗證時將實驗的資料分成五個 fold進行交叉驗證，實驗數據如表2。

表 3 實驗結果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean |
| CT | 0.8888 | 0.8239 | 0.8572 | 0.8322 | 0.8959 | 0.8596 |
| FT | 0.8566 | 0.8063 | 0.8091 | 0.8133 | 0.8604 | 0.8291 |
| MN | 0.5887 | 0.5567 | 0.675 | 0.6638 | 0.6216 | 0.6212 |
| Mean | 0.7781 | 0.7289 | 0.7804 | 0.7698 | 0.7926 | 0.77 |

從實驗結果可以看到的是MN的正確率都比CT與FT還要低上許多，可能是因為單是從影像上用肉眼判斷就不容易將MN從CT中區分開來，MN本身的大小又相對小了許多，就算判別出來，分割的形狀也不一定正確。

# 參考文獻

1. Olaf Ronneberger, Philipp Fischer, Thomas Brox,  
   “U-Net: Convolutional Networks for Biomedical Image Segmentation,”   
   arXiv preprint arXiv:1505.04597, 2015.
2. Jonathan Long, Evan Shelhamer, Trevor Darrell,   
   “Fully Convolutional Networks for Semantic Segmentation,”   
   arXiv preprint arXiv:1411.4038, 2014.
3. Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, Jongyoul Park,   
   “An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection,” arXiv preprint arXiv:1904.09730, 2019.
4. Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang,   
   “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network,” arXiv preprint arXiv:1609.05158, 2016.