國立中正大學工學院機械工程學系

期末專題報告

深度學習運用下顎神經管萃取

指導教授：姚宏宗教授

指導研究生：張家豪 學長

專題生：許福全、劉承祐

中華民國 108 年 1 月 14 日

目錄

研究動機 3
研究目的 3
研究方法 4
程式語言選擇 4
機械學習框架 4
深度學習模型 5
CNN、RNN 比較 5
專題分工 6
介面製作 6
Gui 介面製作 6
使用 Qt 介面製作 7
PyQt5 文件 to Python 文件 7
從介面、檔案、傳遞到顯示 8
3D 建模 9
計算 Dcm 之間的 z 座標距離 9
計算 Dcm 的 CT 值(unit:Hu) 9
Resample 重採樣 10
3D 模型繪製 10
Threshold 閥值 11
執行之後的問題 11
GDCM 12
結論 13
參考資料 13

研究動機

因應牙醫學醫院方面的需求，在下顎齒骨中的神經管需要有一個精確、簡單明瞭的辨識方式。

下顎神經管在齒槽底部，一般而言不容易接觸，但也有需要注意的例外，如：

- 植牙
- 抽神經
- 牙齒重建

在植牙過程中，在植牙過程中，因為需要較深的空間提供植牙，因此往往會需要特別注意下顎神經管位置，有分析神經管醫學影像的需求。

研究目的

- （上學期）

  使用所選擇的程式語言 Python 讀取 Dcm 檔案。

  並於 Gui 介面中進行 2D 影像以及 3D 模型的呈現。

- （下學期）

  令使用者可於影像、模型中進行畫記。

  提供資料庫使程式可藉由深度學習方法，進而自動分析下顎神經管位置。

研究方法

程式語言選擇

本次專題因為主題有 AI 發展相關，故採用 Python 進行實作，選擇使用 Python 的原因有關於其語言特性：

- 學習簡單
- 模組多
- 共享成果多

且 AI 主流的：

- Tensorflow
- PyTorch

都有支援。

機械學習框架

使用的框架是PyTorch，PyTorch原生支援Python，安裝更方便、BUG也更少。

框架名稱	組織	API支援語言	Star	Fork	參考網址
Tensorflow	Google	C++, Python, GO, Java	98,120	62,206	https://github.com/tensorflow/tensorflow
Keras	François Chollet	Python	28,880	10,732	https://github.com/keras-team/keras
Caffe	BVLC	C++, Python, Matlab	23,939	14,649	https://github.com/BVLC/caffe
PyTorch	Adam Paszke	Python	14,692	3,290	https://github.com/pytorch/pytorch
CNTK	Microsoft	C++, C#, Python, Java	14,345	3,819	https://github.com/Microsoft/CNTK
MXNet	DMLC	C++, Scala, R, JS, Python, Julia, Matlab, Go	13,816	5,120	https://github.com/apache/incubator-mxnet
DL4J	DeepLearning4Java	Java, Scala	8,807	4,216	https://github.com/deeplearning4j/deeplearning4j
Theano	University of Montreal	Python	8,175	2,454	https://github.com/Theano/Theano
Torch7	Facebook	Lua	7,866	2,276	https://github.com/torch/torch7
Caffe2	Facebook	C++, Python	7,849	1,914	https://github.com/caffe2/caffe2
Paddle	Baidu(百度)	C++, Python	6,818	1,853	https://github.com/PaddlePaddle/Paddle
DSSTNE	Amazon	C++	4,098	676	https://github.com/amzn/amazon-dsstne
tiny-dnn	tiny-dnn	C++	4,044	1,075	https://github.com/tiny-dnn/tiny-dnn
Chainer	Chainer	Python	3,727	982	https://github.com/chainer/chainer
neon	Nervana Systems	Python	3,476	793	https://github.com/NervanaSystems/neon
ONNX	Microsoft	Python	3,251	392	https://github.com/onnx/onnx
BigDL	Intel	Scala	2,431	548	https://github.com/intel-analytics/BigDL
DyNet	Carnegie Mellon University	C++, Python	2,248	540	https://github.com/clab/dynet
brainstorm	IDSIA	Python	1,275	154	https://github.com/IDSIA/brainstorm
CoreML	Apple	Python	1,032	97	https://github.com/apple/coremltools

如表格所示，目前絕大多數機械學習框架都與 Python 有關。

統計日期：2018／05／02　https://makerpro.cc/wp-content/uploads/2018/05/表格一.png

深度學習模型

目前常見的深度學習模型包含：

- 監督型（如 CNN）
- 時序型（如 RNN／LSTM）
- 增強學習（如 Q-Learning）
- 轉移學習、對抗生成（GAN）

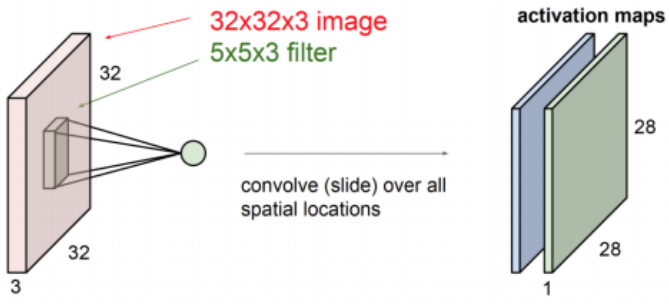
等，但並非每個框架都能全部支援。

本次實驗採用 CNN 模型進行操作，在下一小節也將比對最為人所知的 CNN、RNN 兩種框架的差異。

CNN、RNN 比較

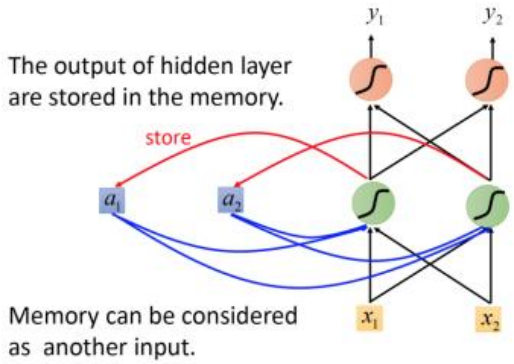
CNN:

每層神經網絡橫向可以多個神經元共存,縱向可以有多層神經網絡連接，善於抽取位置不變特徵。因此在於空間擴展、神經元與特徵卷積等靜態輸出方面表現較優異。



RNN:

RNN 與 CNN 最大的不同在於: RNN 在每一層之間皆有儲存、反饋的存在，類似記憶功能，因此 RNN 對於與時間擴展、前後對比有關的方面表現優異，如: 語言分析。



專題分工

- 許福全：

  大部分著重圖像,針對dcm檔案使用Sitk進行3D化顯示、圖片顯示轉換、使用 pyOpenGL 進行動態操作…等。
- 劉承祐：

  大部分著重介面，並提供圖片串接的函式、使用者操作轉換、檔案之間的分割與傳遞…等。
- 共同：

  numpy、pyplotlib 函式庫使用方法研究。

  針對 Pytorch、CNN 學習，先了解理論進行，以在下學期時能更加快速地進入下一個專題階段。

介面製作

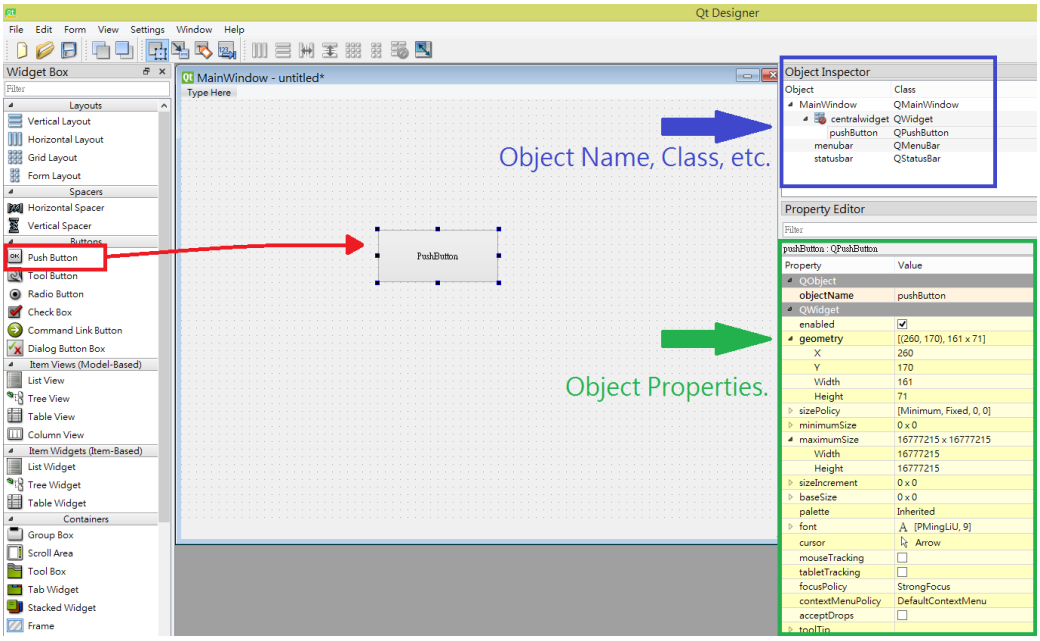
Gui 介面製作

- 介面使用 Python Qt 製作
- 選擇 PyQt 的原因
- 製作上簡單方便
- Qt 發展歷史已廿餘年
- 公開 Qt 程式碼函式庫已近 10 年發展穩定。

使用 Pip 安裝 Python Qt，指令：　pip install PyQt5

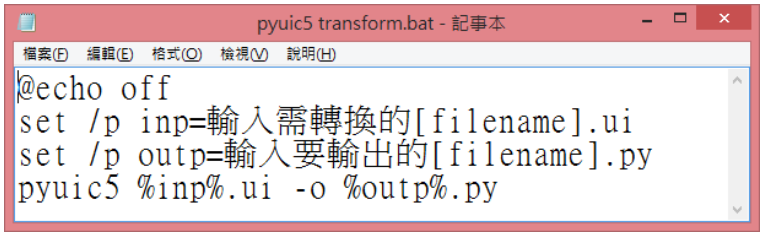
使用 Qt 介面製作

Qt 製作介面時的使用方法非常人性化，僅需拖放，就可以把介面各式所需的功能框框設定完畢。



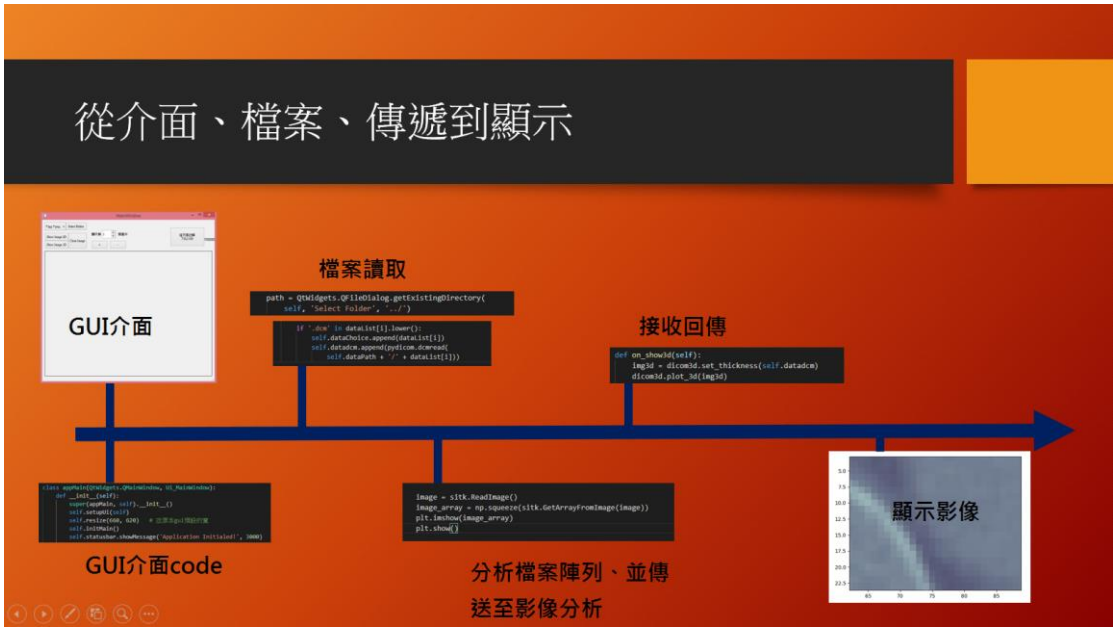
PyQt5 文件 to Python 文件

Qt 文件存檔類型為 xml 格式的 .ui file，與 Python 語法不同，不能共用。因此使 pyuic5 (PyQt5 套件之一)轉換檔案，可以直接生成.py file，其語法即為 Python 使用。




```
1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file 'gui.ui'
4  #
5  # Created by: PyQt5 UI code generator 5.11.3
6  #
7  # WARNING! All changes made in this file will be lost!
8
9  from PyQt5 import QtCore, QtGui, QtWidgets
```

從介面、檔案、傳遞到顯示



GUI 介面:

利用 PyQt 所產生的介面 code，在 Python 環境下執行可以得到所建構的使用者介面。

檔案讀取

利用 Qt 的 QtWidgets.QFileDialog.getExistingDirectory() 函式，可以實現 GUI 版本的選擇資料夾功能，再依資料夾內順序依序讀取檔案。

分析並傳送檔案影像陣列

檔案為影像時，利用 sitk.GetArrayFromImage()取得影像陣列並儲存到 np.squeeze()中，再將儲存的陣列傳送至影像分析部分。

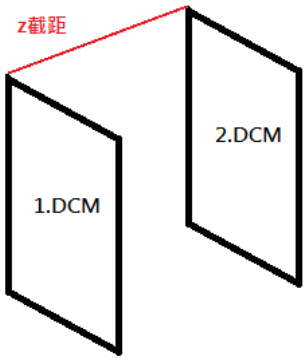
接收回傳資料、顯示影像

利用自製函式 dicom3d.set_thickness()與 dicom3d.plot_3d()實現顯示功能。

3D 建模

計算 Dcm 之間的 z 座標距離

利用 Dcm 檔案內的 Image Position Patient 屬性，
得知圖像左上邊界的位置進而計算 z 截距。



或是利用 Slice Location 屬性來計算。

```
def set_thickness(dcm):
    try:
        slice_thickness = np.abs(dcm[0].ImagePositionPatient[2] - dcm[1].ImagePositionPatient[2])
    except:
        slice_thickness = np.abs(dcm[0].SliceLocation - dcm[1].SliceLocation)
    for s in dcm:
        s.SliceThickness = slice_thickness
    return dcm
```

計算 Dcm 的 CT 值(unit:Hu)

CT 值公式 = 灰階值 * Rescale Slope + Rescale Intercept

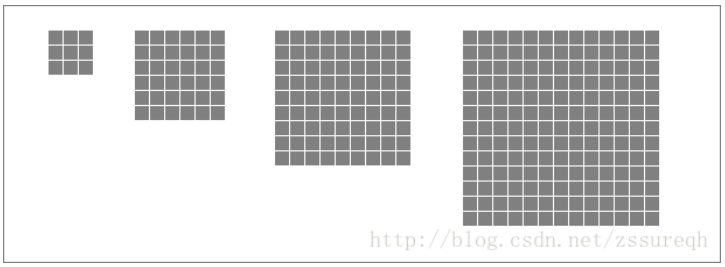
從 pixel_array 屬性取得灰階值，並令空氣(灰階值 =-2000)的灰階值歸零。

取得每一張 Dcm 的縮放斜率和縮放截距，進而計算其 CT 值。

```
def get_pixels_hu(dcm):
    image = np.stack([s.pixel_array for s in dcm])
    image = image.astype(np.int16)
    image[image == -2000] = 0
    for slice_number in range(len(dcm)):
        intercept = dcm[slice_number].RescaleIntercept
        slope = dcm[slice_number].RescaleSlope
        if slope != 1:
            image[slice_number] = slope * image[slice_number].astype(np.float64)
            image[slice_number] = image[slice_number].astype(np.int16)
        image[slice_number] += np.int16(intercept)
    return np.array(image, dtype=np.int16)
```

Resample 重採樣

由第一張 Dcm 的像素尺寸去重新設定其他 Dcm 的像素尺寸。



利用 scipy.ndimage.interpolation.zoom 函數，將前面計算的 CT 值進行陣列大小縮放。

```
def resample(dcm_hu, dcm, new_spacing=[1, 1, 1]):
    spacing = map(float, (dcm[0].SliceThickness] +dcm[0].PixelSpacing))
    spacing = np.array(list(spacing))
    resize_factor = spacing / new_spacing
    new_real_shape = dcm_hu.shape * resize_factor
    new_shape = np.round(new_real_shape)
    real_resize_factor = new_shape / image.shape
    new_spacing = spacing / real_resize_factor
    dcm_hu = scipy.ndimage.interpolation.zoom(
        dcm_hu, real_resize_factor, mode='nearest')
    return dcm_hu, new_spacing
```

3D 模型繪製

將圖像的座標(x,y,z)轉成(z,y,x)，令模型豎立以方便觀察。

利用 marching_cubez_lewiner 函數，將 CT 值與 threshold 變數進行比對；若相同，則將其顯示於立方體內。

```
def plot_3d(image, threshold=-300):
    pic = image.transpose(2, 1, 0)
    verts, faces = measure.marching_cubes_lewiner(pic, threshold)
    fig = plt.figure(figsize=(10, 10))
    ax = fig.add_subplot(111, projection='3d')
    mesh = Poly3DCollection(verts[faces], alpha=0.1)
    face_color = [0.5, 0.5, 1]
    mesh.set_facecolor(face_color)
    ax.add_collection3d(mesh)
    ax.set_xlim(0, pic.shape[0])
    ax.set_ylim(0, pic.shape[1])
    ax.set_zlim(0, pic.shape[2])
    plt.show()
```

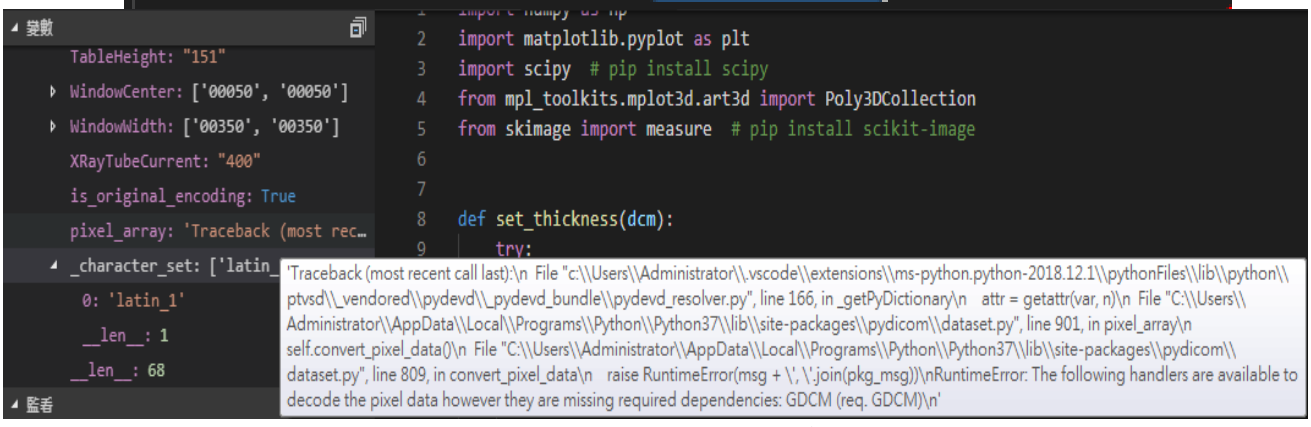
Threshold 閥值

正常人体组织的CT值（HU）

组织	平均CT值	组织	平均CT值
脑	25~45	肌肉	35~50
灰质	35~60	淋巴结	45+/-10
白质	25~38	脂肪	-80~-120
基底节	30~45	前列腺	30~75
脑室	0~12	骨头	150~1000
肺	-500~-900	椎间盘	50~110
甲状腺	100+/-10	子宫	40~80
肝	40~70	精囊	30~75
脾	50~70	水	0
胰腺	40~60	空气	-1000
肾	40~60	静脉血液	55+/-5
主动脉	35~50	凝固血液	80+/-10

執行之後的問題

```
def get_pixels_hu(dcm):
    image = np.stack([s.pixel_array for s in dcm])
```



pixel_array 變數讀取失敗，回報需求 GDCM 的函式庫。

GDCM

Supported Transfer Syntaxes

As far as we have been able to verify, the following transfer syntaxes are handled by the given packages:

Transfer Syntax					
Name	UID	NumPy	NumPy + JPEG-LS	NumPy + GDCM	NumPy + Pil
Explicit VR Little Endian	1.2.840.10008.1.2.1	✓	✓	✓	✓
Implicit VR Little Endian	1.2.840.10008.1.2	✓	✓	✓	✓
Explicit VR Big Endian	1.2.840.10008.1.2.2	✓	✓	✓	✓
Deflated Explicit VR Little Endian	1.2.840.10008.1.2.1.99	✓	✓	✓	✓
RLE Lossless	1.2.840.10008.1.2.5	✓	✓	✓	✓
JPEG Baseline (Process 1)	1.2.840.10008.1.2.4.50			✓	✓[1]
JPEG Extended (Process 2 and 4)	1.2.840.10008.1.2.4.51			✓	✓[1]
JPEG Lossless (Process 14)	1.2.840.10008.1.2.4.57			✓	
JPEG Lossless (Process 14, SV1)	1.2.840.10008.1.2.4.70			✓	✓
JPEG LS Lossless	1.2.840.10008.1.2.4.80		✓	✓	
JPEG LS Lossy [3]	1.2.840.10008.1.2.4.81		✓	✓	
JPEG2000 Lossless	1.2.840.10008.1.2.4.90			✓	✓[2]
JPEG2000 [4]	1.2.840.10008.1.2.4.91				✓[5]
JPEG2000 Multi-component Lossless	1.2.840.10008.1.2.4.92				

Grassroots Dicom:

具有 Dicom 的文件格式定義和網路通訊協定，可以協助讀取 Dicom 檔案。

結論

深度學習:

深度學習中使用到的深度神經網路其實不是這幾年才出現的新東西，大多數理論基礎都是在 10 年或更久之前就被開發出來了。然而受限於當時電腦的運算能力和數位資料不足，沒辦法訓練出夠好的類神經網路模型。

當運算能力和資料不再是門檻之後，深度學習也更迅速地融入我們的生活之中，成為大眾可以運用的新技術。

當我們贊嘆深度學習的強大之時，也要注意深度學習的本質和侷限，才能正確的有效的運用這個強大的技術。也讓我們一起期待更多更有趣的深度學習應用。

研究成果:

目前不論是介面、圖像方面都遇到了相同的難點: GDCM 函式庫無法正常使用。

GDCM 函式庫雖然於作者 Github 有下載 Windows 系統支援的安裝包，但安裝後卻仍然異常、無法執行。

估計於下學期會改為使用 Linux Like 作業系統進行操作，將再測試 Linux 版本的安裝包安裝後是否可以正常使用。

參考資料

QT 歷史:https://zh.wikipedia.org/wiki/Qt#歷史
CNN.RNN 差異:https://hk.saowen.com/a/b52c246a27d0990a23211e24242d2c36b946488ce972276eb833c1910f808a7b
深度學習:
http://www.cc.ntu.edu.tw/chinese/epaper/0038/20160920_3805.html
GDCM Github: https://github.com/malaterre/GDCM/releases
Dicom 各個 Tag 屬性: https://www.cnblogs.com/stephen2014/p/4579443.html
3D 模型範例: http://shartoo.github.io/medical_image_process/
Resample: https://blog.csdn.net/zssureqh/article/details/78768942
Why needs GDCM:
https://pydicom.github.io/pydicom/stable/image_data_handlers.html