# 4.1

加油！只要用心努力，总有一天会有所回报。念念不忘，必有回响。

生活在这个世界上，没有一刻不是苦恼的，但也因此会有快乐，好好享受生活，就要拥抱，不管是痛苦，还是幸福。没有痛苦就不会有幸福，没有幸福就不会有痛苦。

我希望你能成为一个总是向前看，拥有积极向上的心态，不折不挠的精神，无论遇到什么样的挫折与不公，最后总能再次振作起来的人。

要为自己做的事情负责，逃避无法解决问题，就算一直逃避下去，问题还是在那里。

|  |
| --- |
| python学习： |
| **1.列表：**  例如：bicycles=[‘a’,’b’,’c’]  print(bicycles[0]) //a  ·索引:从0开始，-1表示返回最后一个元素，-2，-3以此类推  ·添加元素：.append()  ·插入元素：.insert(0,’d’) //[‘d’,’a’,’b’,’c’]  ·删除元素：①不使用删除的元素：del bicycles[0];  ②使用删除的元素：last\_bicycle=bicycles.pop() //pop(3)删除索引为3的元素  ③根据值删除元素：bicycles.remove(‘a’); //只能删除第一个指定的元素  ·组织列表：①永久性排序：bicycles.sort();//sort(reverse=True)与字母顺序反序  ②临时性排序：sorted(bicycles)  ③倒着打印列表：bicycles.reverse()  ④确定列表长度：len(bicycles)  ·操作列表：   1. for循环遍历列表，缩进代表要进行for循环的语句  |  | | --- | | for bicycle in bicycles **:**  print(bicycle) | | 输出：  a  b  c |  1. 数字列表   range()使用：   |  | | --- | | for value in range(1,5):  print(value)//实际打印的是1-4，左闭右开 | | 用list函数直接转换为数字列表  numbers=list(range(1,5))  print(numbers) //[1,2,3,4]  numbers=list(range(2,11,2)) //指定步长  print(numbers) //[2,4,6,8,10]  squares=[]  for number in range(1,11)  squares=.append(number\*\*2);//将前十个数平方加入一个列表中  列表解析：squares=[value\*\*2 for value in range(1,11)]  for number in range(30) : //这样可以用来控制循环次数 |   统计计算：  min(numbers)  max(numbers)  sum(numbers)   1. 列表切片  |  | | --- | | bicycles=[‘a’,’b’,’c’,’d’]  print(bicycles[0:3]) //也是左闭右开：[‘a’,’b’,’c’]  print(bicycles[2:]) //[c,d]  print(bicycles[:3]) //[a,b,c]  print(bicycles[-3:]) //输出最后三个元素[b,c,d] | | 遍历切片：  for bi in bi[0:3] : |  1. 复制列表   用切片：mines=friends[:] //不能使用mines=friends这样相当于共用了一个列表   1. 元组tuple：元素不可变的列表，用圆括号标识   例如：dimensions=(250,50)  dimensions[0]=200会报错，因为元组元素不可变  dimensions=(200,100)这样不会报错，可以这样改 |
| **2.if语句**   |  | | --- | | for car in cars :  if car = ‘bwm’ :  print(car.upper())  else :  print(car.title()) | | 条件1 and/or 条件2 | | 判断特定值是否在列表中：if 元素 (not) in 列表 | | if elif else语句 | | 判断列表是否为空  if requests : //若不为空则为true | |
| **3.字典**  例：family={ ‘mom’: ’lxy’ , ‘mom\_year’ : 48}可以存储一个对象的多种信息，也可以存储多个对象的一种信息。  ·访问：family[ ‘mom’ ]  ·添加/修改：直接赋值  ·删除键值对：del family[ ‘mom’ ]  ·遍历字典：   |  | | --- | | ①遍历所有键值对：用items()方法  for key,value in family.items() :  ②遍历所有键：用keys()方法  for key in family.keys() :  ③顺序遍历字典中所有键：  for key in sorted(family.keys()) :  ④遍历所有值values()  for key in family.values() ://若要不重复的值：set(family.values()) |   ·嵌套：  ①将字典嵌套在列表里，比如说  字典user\_1、user\_2、user\_3包含一个用户的各种信息  列表users:users=[user\_1, user\_2, user\_3]  ②在字典中存储字典，即字典作为键值对的值 |

# **[医学图像之DICOM格式解析](https://www.cnblogs.com/XDU-Lakers/p/9863114.html)**

<https://www.cnblogs.com/XDU-Lakers/p/9863114.html>

# DICOM文件下载

<https://blog.csdn.net/calmreason/article/details/91047127>

# vtk读取医疗图像、裁剪、提取等操作的学习记录

[https://blog.csdn.net/Always\_ease/article/details/88844187?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522158549820419724835834623%2522%252C%2522scm%2522%253A%252220140713.130056874..%2522%257D&request\_id=158549820419724835834623&biz\_id=0&utm\_source=distribute.pc\_search\_result.none-task](https://blog.csdn.net/Always_ease/article/details/88844187?ops_request_misc=%7B%22request%5Fid%22%3A%22158549820419724835834623%22%2C%22scm%22%3A%2220140713.130056874..%22%7D&request_id=158549820419724835834623&biz_id=0&utm_source=distribute.pc_search_result.none-task)

simpleITK使用：

关键一步：读出像素矩阵

pixel\_array = sitk.GetArrayFromImage(file)#这个file是之前读取出来的文件

例子：

# 4.2

1. pixel\_array不能直接用，要处理为pixel.data

|  |
| --- |
| 如果要对dicom图像中的像素值进行修改，继续执行以下代码：  ##修改图片中的元素，不能直接使用data\_array,需要转换成PixelData  for n,val in enumerate(ds.pixel\_array.flat): # example: zero anything < 300  if val < 300:  ds.pixel\_array.flat[n]=0  ds.PixelData = ds.pixel\_array.tostring()  ds.save\_as("newfilename.dcm")  ————————————————  版权声明：本文为CSDN博主「愿十四亿神州尽舜尧」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。  原文链接：https://blog.csdn.net/weixin\_41783077/article/details/82990050 |

**今天学到了/解决了什么问题：**

|  |
| --- |
| 1. 关于dicom文件的读取：<https://www.cnblogs.com/jxblog/p/12010354.html>   ①使用pydicom：  ds = pydicom.read\_file  #我们需要的对象主要是ds.pixel\_array  ②使用simpleITK  读取单张：  image = sitk.ReadImage(slice\_path) image\_array = sitk.GetArrayFromImage(image) # z, y, x np\_array = np.int16(image\_array) image\_array = np.squeeze(np\_array)  读取序列：  reader = sitk.ImageSeriesReader() case\_path="E:\\dicomdata\data\data4\\3b\_mpr\_pr\_hf\_vfh" #case\_path="E:\dicomdata\data\data1\series-000001" dicom\_names = reader.GetGDCMSeriesFileNames(case\_path) reader.SetFileNames(dicom\_names) image = reader.Execute() image\_array = sitk.GetArrayFromImage(image) # z, y, x origin = image.GetOrigin() # x, y, z origin是指实际像素大小 spacing = image.GetSpacing() # x, y, z  image是一个三维矩阵，可以像读取一张切片一样，读取它的原点，像素间隔，方向，等基本信息   1. 显示dicom文件：   pylab.imshow(ds.pixel\_array, cmap=pylab.cm.bone) pylab.show()   1. 可以将dicom序列转化成nii数据   好处是python中有一个库nibabel专门处理nii文件  nii文件是三维的，而dicom是二维的多张图像  4.dicom文件有单张的，也有多张的序列。可以通过用sitk来读取dicom文件的序列号来查看是不是同一个序列。 |

**仍未解决的问题：**

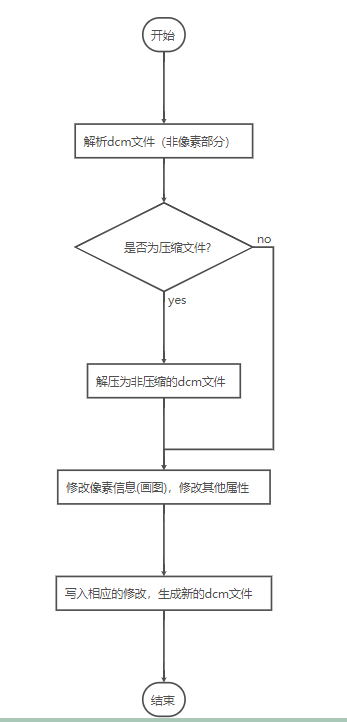
|  |
| --- |
| 1. 怎么显示？dicom序列 2. 想要将dicom序列合并成一个dicom文件，但是提示错误 3. GDCMImageIO(0000015DFD571CF0): DICOM does not support this component type 4. 是否需要做一个可视化的操作？ 5. 有没有读取dicom数据的方便的工具 |

**dicom插值**

|  |
| --- |
| 处理文件 处理DICOM文件主要有插值等操作，可以直接使用SimpleITK（或者说是ITK）的相关函数，并通过pipeline结构进行处理。 插值 import SimpleITK as sitk  reader = sitk.ImageSeriesReader()  dicom\_names = reader.GetGDCMSeriesFileNames(case\_path) reader.SetFileNames(dicom\_names)  image = reader.Execute()  resample = sitk.ResampleImageFilter()  resample.SetOutputDirection(image.GetDirection())  resample.SetOutputOrigin(image.GetOrigin())  newspacing = [1, 1, 1]  resample.SetOutputSpacing(newspacing)  newimage = resample.Execute(image)  作者：张杰Phantom 链接：https://www.jianshu.com/p/deab28f0cef7 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。 |

# Pydicom+SimpleITK操作DICOM图像数据和TAG

<https://blog.csdn.net/weixin_42663215/article/details/90899819?depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-16&utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-16>



**明天可以看一下：**

<https://blog.csdn.net/qq_31325495/article/details/82015784?utm_source=blogxgwz0>

# **医疗图像处理基础知识与常用工具**

<https://www.dazhuanlan.com/2019/10/14/5da44ff5b8589/>

**python用pydicom读取dicom文件详解**

<https://jingyan.baidu.com/article/0eb457e5e8177303f1a905d2.html>

**关于图片配准**

<https://blog.csdn.net/gaoaoysjl/article/details/91541955?depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-10&utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-10>

# 4.3

开会问题:

1. 本课题的进度安排如下：
2. 代码量起码3000行
3. 重点是配准

本周任务：

可以把dicom数据转化jpg，png图片，

读取图片，应用文献里的公式。

# 4.4

学姐发来的代码：

0.学习opencv用法

1.学习读取dicom文件以及阈值分割，这里的

dcm.image = dcm.pixel\_array \* dcm.RescaleSlope + dcm.RescaleIntercept

 #有的CT图像中像素不是HU值，Rescale Intercept和Rescale Slope这两个用于将像素转化为HU值

1. 显示dicom文件，为什么要用0.0006785\*pixel\_array
2. 读取jpg图像获得图片的宽度长度像素值等
3. 4里的算法是什么，对两幅dcm图片的pixel\_array进行处理，产生了一张png图片
4. 使用tkinter进行交互
5. 鼠标点击绘制roi区域
6. roi区域

8和9处理pixel\_array

pyplot学习：

plt.subplot(131)绘子图：意思是一行三列，这是第一列

1：行数；3：列数；1：索引

明天要完成的事情：

1. 读学姐发来的英文文献 ok

多参数MRI核磁共振检查

bvf血容量分数

1. 读懂 截取roi区域的代码 no
2. 学习tkinter no

# 4.5

今天读了一下那篇文献，顺便问了一下学姐问题

那个公式主要是应用于脑肿瘤。

从一个网站上下载了有无脑肿瘤的数据集。但是不是dcm格式，可以进行转化。

初步可以进行这样的设想：

1. 将用户上传的dicom文件进行数据提取：患者姓名，等。并且转化为png格式。
2. 对上传的/转化完成的png图片进行处理分析
3. 首先对图片进行预处理，归一化等等
4. 对图片进行配准
5. 手动绘制roi区域
6. 对roi区域进行叠加，计算像素值，判断可能出现异常的部位。

# 4.6

1.上采样： 放大图像（或称为上采样（upsampling）或图像插值（interpolating））

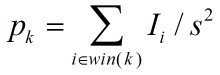
·主要目的是放大原图像,从而可以显示在更高分辨率的显示设备上。对图像的缩放操作并不能带来更多关于该图像的信息, 因此图像的质量将不可避免地受到影响。然而，确实有一些缩放方法能够增加图像的信息，从而使得缩放后的图像质量超过原图质量的。

·上采样原理：图像放大几乎都是采用内插值方法，即在原有图像像素的基础上在像素点之间采用合适的插值算法插入新的元素。

2.下采样：downsampling

·主要目的有两个：1、使得图像符合显示区域的大小；2、生成对应图像的缩略图。

·下采样原理：对于一幅图像I尺寸为M\*N，对其进行s倍下采样，即得到(M/s)\*(N/s)尺寸的得分辨率图像，当然s应该是M和N的公约数才行，如果考虑的是矩阵形式的图像，就是把原始图像s\*s窗口内的图像变成一个像素，这个像素点的值就是窗口内所有像素的均值：



3.插值算法分类

对插值算法分类比较混乱，各人有各人的分类算法。文献《图像插值技术综述》中简略的将插值算法分为传统插值、 基于边缘的插值和基于区域的插值3类，作为初学者入门明晰插值算法还是有帮助。

1.传统差值原理和评价

在传统图像插值算法中，邻插值较简单，容易实现，早期的时候应用比较普遍。但是，该方法会在新图像中产生明显的锯齿边缘和马赛克现象。双线性插值法具有平滑功能，能有效地克服邻法的不足，但会退化图像的高频部分，使图像细节变模糊。在放大倍数比较高时，高阶插值，如双三次和三次样条插值等比低阶插值效果好。这些插值算法可以使插值生成的像素灰度值延续原图像灰度变化的连续性，从而使放大图像浓淡变化自然平滑。但是在图像中，有些像素与相邻像素间灰度值存在突变，即存在灰度不连续性。这些具有灰度值突变的像素就是图像中描述对象的轮廓或纹理图像的边缘像素。在图像放大中，对这些具有不连续灰度特性的像素，如果采用常规的插值算法生成新增加的像素，势必会使放大图像的轮廓和纹理模糊，降低图像质量。

2.基于边缘的图像插值算法

为了克服传统方法的不足， 提出了许多边缘保护的插值方法，对插值图像的边缘有一定的增强， 使得图像的视觉效果更好， 边缘保护的插值方法可以分为两类： 基于原始低分辨图像边缘的方法和基于插值后高分辨率图像边缘的方法。基于原始低分辨率图像边缘的方法:( 1)首先检测低分辨率图像的边缘， 然后根据检测的边缘将像素分类处理， 对于平坦区域的像素，采用传统方法插值；对于边缘区域的像素， 设计特殊插值方法， 以达到保持边缘细节的目的。(2)基于插值后高分辨率图像边缘的方法这类插值方法:首先采用传统方法插值低分辨率图像，然后检测高分辨率图像的边缘，最后对边缘及附近像素进行特殊处理， 以去除模糊， 增强图像的边缘。

3.基于区域的图像插值算法

首先将原始低分辨率图像分割成不同区域，然后将插值点映射到低分辨率图像， 判断其所属区域， 最后根据插值点的邻域像素设计不同的插值公式， 计算插值点的值。

1. 图像的灰度值转化为CT值即HU值

<https://blog.csdn.net/m0_37477175/article/details/103803321?depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1&utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1>

用从DICOM文件中读取的rescaleslope值和rescale intercept值可以将每个像素的灰度值转换成CT值：**Hu=pixel\_val\*rescale\_slope+rescale\_intercept;**

pixel\_val是第i个像素的灰度值

Hu是第i个像素的CT值

原文链接：https://blog.csdn.net/yunzhaoqiang/article/details/62215922

1. 全局阈值分割

一般图像分割是基于图像二值化的基础上进行的，其目的是可减少图像的灰度分布范围，简化运算，从而大大提高分割速度。

但如遇到特殊情况，例如脑部的头骨，高密度，灰度值高，处于亮区域，而我们感兴趣的区域（肿瘤）的灰度是软组织，灰度值低，处于暗区域。而肿瘤周围的软组织也是灰度值低的暗区域，与肿瘤的区别很小。此时，原始方法已不再能分辨。所以在程序中，必须舍弃。现有的二值化自动分割函数，改为手动的全局阈值分割。通过观察，设置特定的阈值点，从而进行有效手动分割，将处于亮区域的头骨和其他组织分割开，又保留了其他组织的原有特性。将其作为预处理图像

1. 肿瘤分割

经过手动的全局阈值分割后的图片，虽然消除了头骨的部分，但由于我们感兴趣的部分（肿瘤）并没有从周围组织里面分割出来，因此还需要进一步的处理。由于肿瘤和周围的组织灰度值比较接近，我们分别选用最大方差阈值法和形态学两种方法来处理。

3.3.1最大方差阈值法分割

最大方差阈值也叫大津阈值，他是在差别域最小二乘法原理的基础上推导出来的，不管是图像的直方图有无明显的双峰，此方法都能取得很好的效果。其原理是把直方图在某一阈值处分割成为两组，当被分成的两组间的方差为最大时，决定阈值。

3.3.2形态学分割

区域生长的基本思想是将具有相似性质的像素结合起来构成区域，具体是先对每个需要分割的区域找一个种子像素作为生长起点，然后将种子像素周围区域中与种子像素有相同或相似性质的像素种子合并到种子像素所在区域中，将这些新像素当作种子像素继续进行上面的过程，直到再也没有满足条件的像素可以被包括进来，这样，一个区域就生成了。经实验显示，采用形态学进行分割之后，脑部的肿瘤可以清晰地显示出来，但与此同时一些血管同时也显示出来了。

具体的算法步骤为：

求出图像的梯度；

采用圆形结构元素执行形态学开操作运算；

进行腐蚀运算；

采用圆形结构元素执行形态学闭运算操作；

进行膨胀运算；

经过开闭运算后，图像进行重建；

将重建的图像叠加在原有图像上。

————————————————

版权声明：本文为CSDN博主「333。0.0」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_40980917/article/details/84551632

原文链接：https://blog.csdn.net/qq\_40980917/article/details/84551632

# 4.10

进度：

1. 难找合适的dicom数据，但是找到了合适的png格式的脑肿瘤数据。准备做一个判断逻辑，用户可以上传dicom格式也可以上传png格式。dicom数据可以获得患者的信息，但需要转化成png图像再进行图像处理。
2. 完成了归一化，下采样，上采样涉及插值算法还在学习
3. 应用了BVf公式
4. 做了一个tkinter可视化界面

目前的设想：

1. 将用户上传的dicom文件进行数据提取：患者姓名，等。并且转化为png格式。
2. 对上传的/转化完成的png图片进行处理分析
3. 首先对图片进行预处理：重采样，归一化等等
4. 对图片进行数据处理，将像素值转化为CT值即HU值，
5. 对图片进行配准
6. 手动绘制roi区域
7. 对roi区域进行校正，叠加，计算像素值，判断可能出现异常的部位。

**下周的问题：**

**校正**，做完之后，接着做一部分人工智能，深度学习。

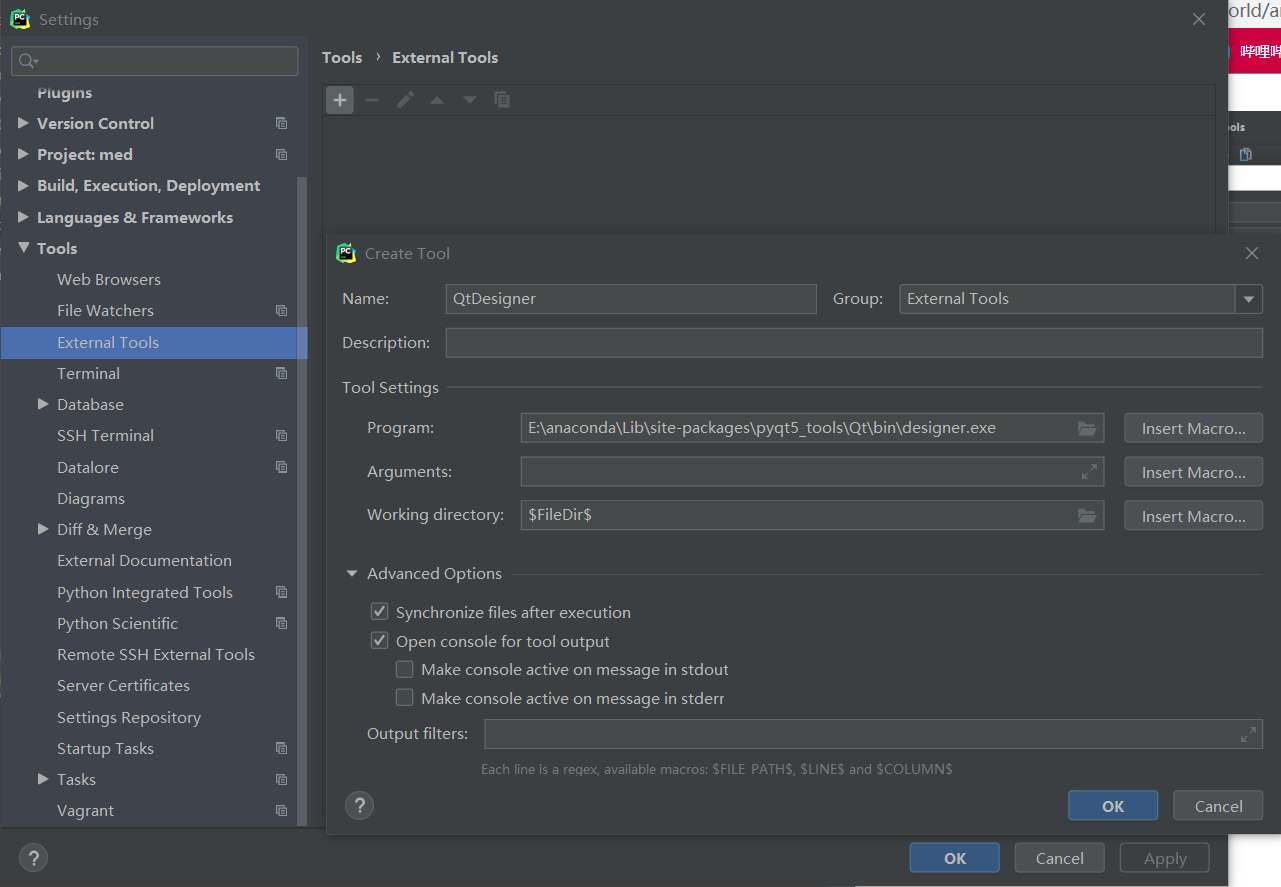
学习资料：乐高积木堆叠强化学习

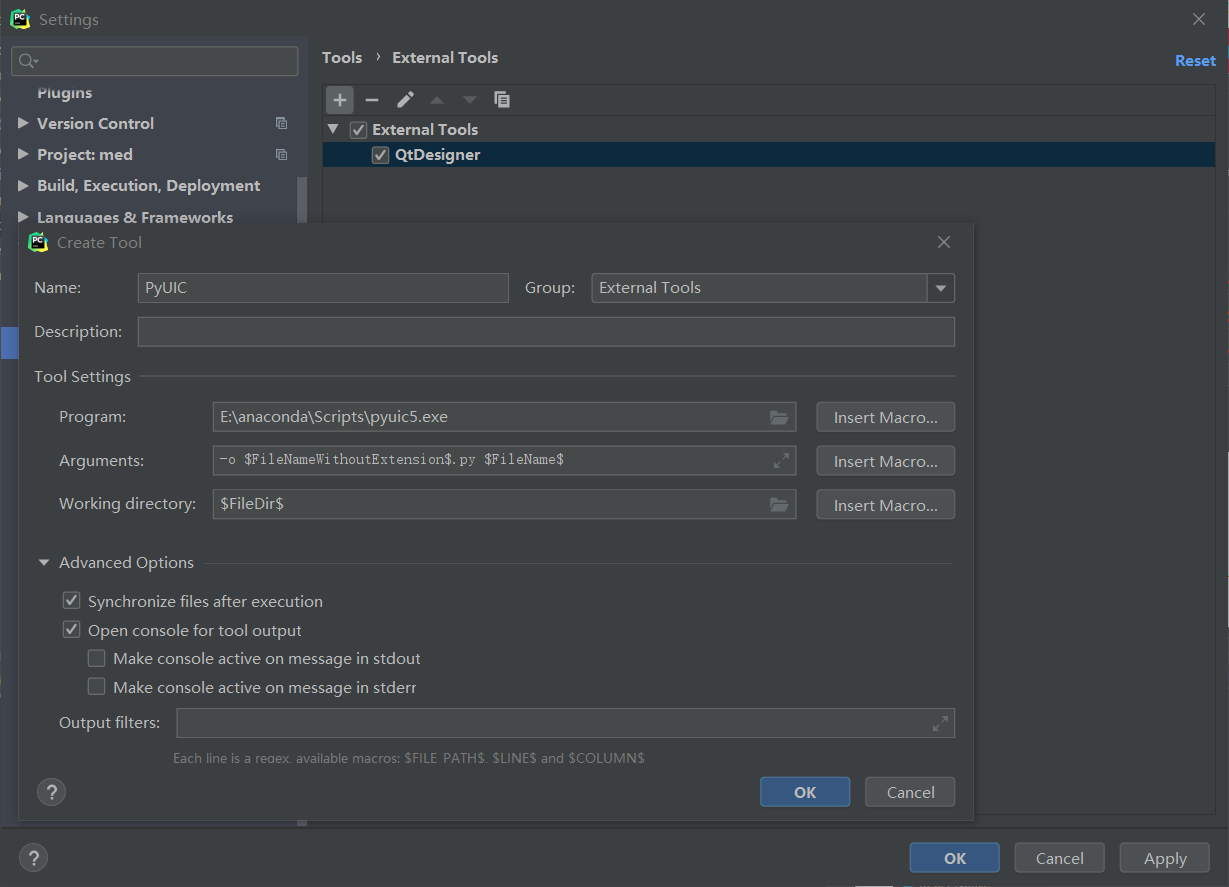
# 4.11

安装工具qypt5

<https://www.cnblogs.com/smarterWorld/articles/11079533.html>

1. pip install pyqt5
2. 安装PyQt5-tools：pip install pyqt5-tools
3. pycharm配置：
   1. 添加external Tools：打开settings->Tools->External Tools点击“+”
   2. 添加QtDesigner
   3. 添加PyUIC





* 1. 项目右键->External Tools”选中打开QtDesigner

1. QT Designer使用
   1. widget box:
      1. spacer:分割器。在设计界面里是蓝色弹簧，其实显示的时候是隐形的
   2. 预览：ctrl+r
2. 好处：将UI（main.py/main.ui）跟逻辑（med.py）分离，因此直接重复步骤7-8即可完成UI的更新，无需改动逻辑（main.py）部分。
3. matplotlib嵌入pyqt5

<https://blog.csdn.net/panrenlong/article/details/80183519>

1. 想了想太麻烦了，干脆先保存图片，再打开显示

# 基于python3.6+pyQT5利用Graphics View 控件显示图像并实现其缩放

<https://blog.csdn.net/weixin_39964552/article/details/82937144>

1. 按键事件：ui.openBTN.clicked.connect(open\_click)
2. 打开子窗口

|  |
| --- |
| app = QApplication(sys.argv)  main = Main()  ch = Child()  main.show()  **main.pushButton.clicked.connect(ch.OPEN)**  sys.exit(app.exec\_()) |

# 4.12

关于预处理：

<https://www.kaggle.com/gzuidhof/full-preprocessing-tutorial>

关于机器学习的一些入门:

<https://blog.csdn.net/binbigdata/article/details/80029734>

# 4.24

周报

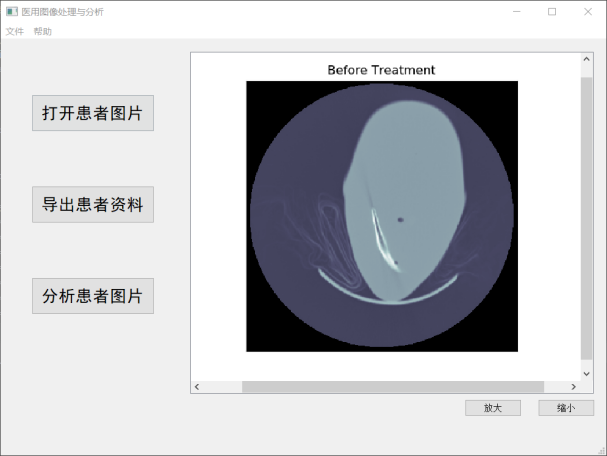
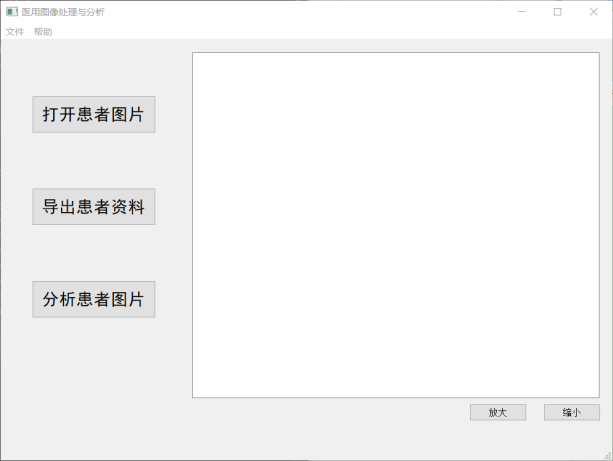
|  |  |  |
| --- | --- | --- |
| 日期 | 学生姓名 | 本周主要做的事情和解决问题，下周要解决的事情 |
| 2020.04.24 | 章寒捷 | 本周：图像配准  下周：学习卷积神经网络CNN并应用 |

1. **工作内容**

图像配准，应用机器学习

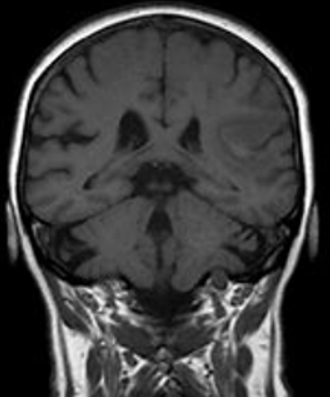
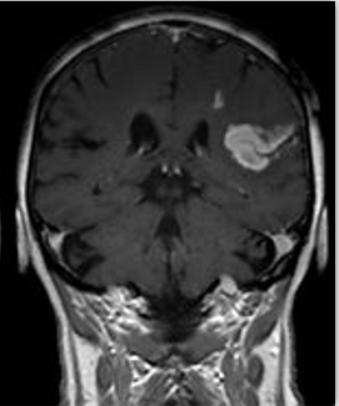
1. **本周具体内容**

1.用pyqt5做了简单的页面设计

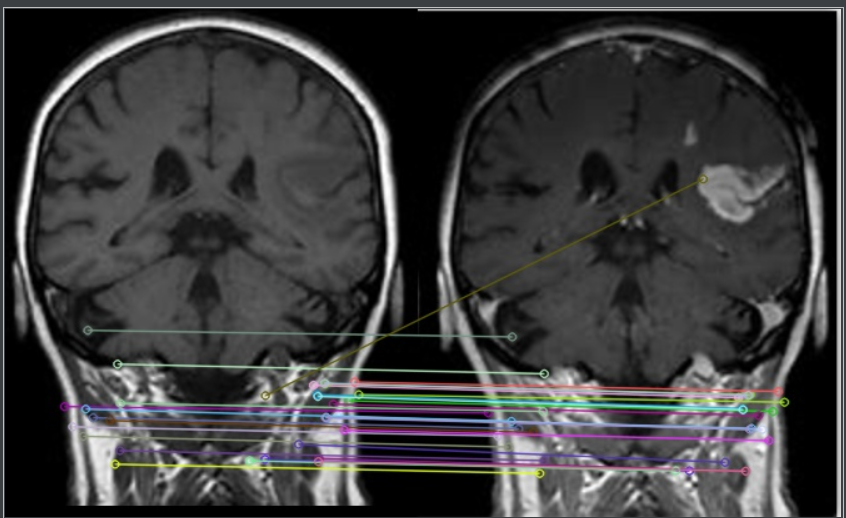


1. 完成了图像预处理：
   1. 采样
   2. 归一化
   3. 将像素值转化为CT值即HU值
2. 这块进度有点慢……找了一些图片配准的资料看，但是很多不适用，代码实现还有很多问题……现在是这样，还没有叠起来

原图：

配准：



1. **下周工作内容**
2. 实现图片配准
3. 有时间再看看机器学习

# 4.24

图片配准：无监督学习的图像非刚性配准模型

1. 什么是图片配准？

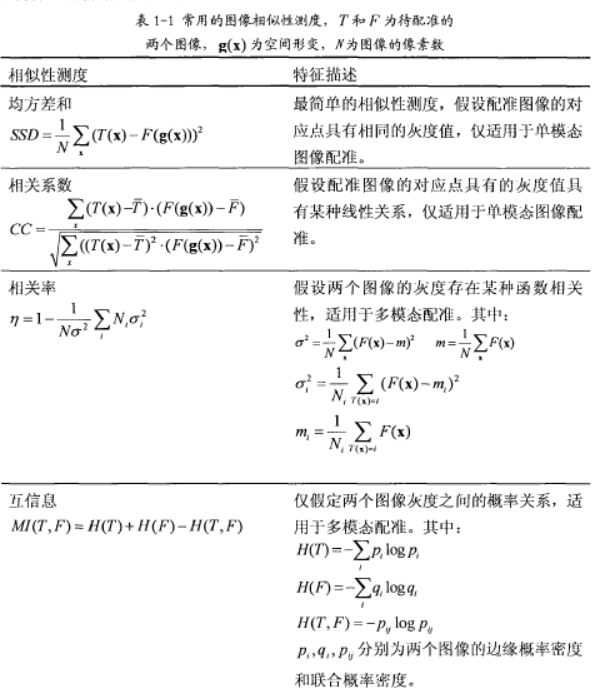
就是对要进行配准的图片寻找一个空间变换，使图片中的匹配点可以互相对齐，从而进行比较或者融合。[1]

1. 图片配准的过程
   1. 用akaze探测器找到特征点
   2. 用bfMatcher暴力匹配找到匹配点，绘制匹配对
   3. 变换，缩放移动
2. 胡永祥. 基于互信息的多模态医学图像非刚性配准研究[D].中南大学,2012.

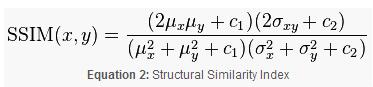
# 4.26

相似度测量：均方差和，相关系数，相关率，互信息

形变模型



现在有一个新的衡量两幅图像结构相似度的新指标：SSIM（structural similarity index measurement），值越大说明越相似，最大为1。



明天：做一个功能，调整图片的亮度和对比度

# 4.26

pip install xxxx -i <https://pypi.douban.com/simple>

进行对比时两张图片大小不一致怎么把

先对原始图片做处理，使其和配准后的图片大小一致，<https://blog.csdn.net/sinat_38814578/article/details/83184742>

这样才能对对其应用ssim测量