

生物医学图像处理实验 2

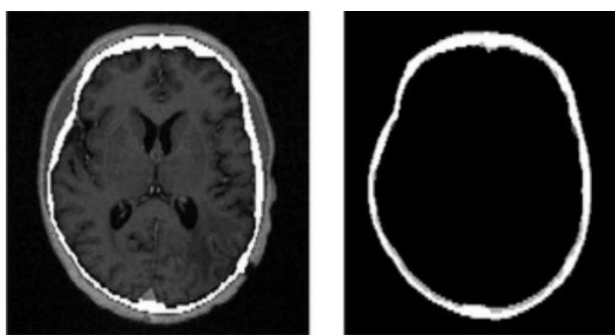
实验目的：

- 1) 理解简单和组合形态学操作的含义及掌握其函数使用
- 2) 使用形态学处理函数实现二值图像和灰度图像的形态学处理

任务：

读取CANVAS系统中的“`OneSlice.dat`”，该数据含有 512×384 像素点数据类型为‘Unit16’的图像，通过形态学处理实现如下：

- 1) 提取边界并与梯度空间滤波结果进行对比。
- 2) 进行图像增强（灰度级形态学处理）
- 3) 分割头骨（例如，边界阈值处理后进行形态学处理，头骨介于脑组织与头皮之间）



要求（任选一种）：

传统方法：如不使用 ChatGPT 等大模型，使用 MATLAB，Python 编程。

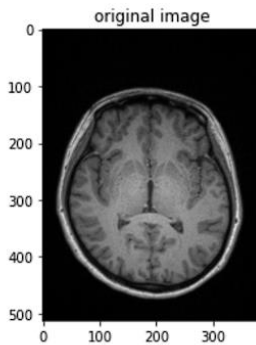
调用大模型方法： 需要实现一个支持交互的界面，根据方法选择， 输出各种结果图像。

请于下周四前提交实验报告（使用传统方法的同学按照之前作业要求方式提交规范报告， 使用大模型方法的同学请记录与大模型交互对话并录制一段交互界面演示图像增强的效果）。

Reading the binary file in Python

<https://github.com/gzr2017/ImageProcessing100Wen>

```
# Read dat file according the given information
im0=np.fromfile('OneSlice.dat',dtype='uint16')
im0.shape=(384,512)
im0=im0.T # Radioloist perferred view
fig=plt.figure()
plt.imshow(abs(im0),cmap='gray') #显示原始的图
plt.title('original image');
```



Reading the binary file in Matlab

```
% Read-in image
fid = fopen('OneSlice.dat');
im = double(fread(fid,[512,384],'uint16'));
fclose(fid);
```

MATLAB: bwmorph

- `BW2 = bwmorph(BW1,OPERATION,N)`
- `OPERATION` is a string that can have one of these values:
 - 'bothat' Subtract the input image from its closing
 - 'branchpoints' Find branch points of skeleton
 - 'bridge' Bridge previously unconnected pixels
 - 'clean' Remove isolated pixels (1's surrounded by 0's)
 - 'close' Perform binary closure (dilation followed by erosion)
 - 'diag' Diagonal fill to eliminate 8-connectivity of background
 - 'endpoints' Find end points of skeleton
 - ...

Python: opencv

```
void cv::morphologyEx(
    cv::InputArray src,
    cv::OutputArray dst,
    int op,
    cv::InputArray element,
    cv::Point anchor = cv::Point(-1,-1),
    int iterations = 1,
    int borderType = cv::BORDER_DEFAULT,
    const cv::Scalar& borderValue = cv::morphologyDefaultBorderValue())
```

`cv2.getStructuringElement`

`cv2.erode`

`cv2.dilate`

`cv2.morphologyEx`

Strel

imdilate

imerode

imclose

imopen

bwlabel

bwmorph

类型	运算	公式	说明
形态学常用算法	边界提取	$\beta(A) = A - (A \ominus B)$	集合A中的边界集合
	孔洞填充	$X_k = (X_{k-1} \oplus B) \cap A^c$ $X_k = X_{k-1}$ 时, 迭代停止	填充A中的空洞。X0的大小与I 相同, 在一个或者多个孔洞中至少有一个点元素为1, 其余为0
	连通分量提取	$X_k = (X_{k-1} \oplus B) \cap I$ $X_k = X_{k-1}$ 时, 迭代停止	寻找I中的连通分量。X0的大小与I相同, 在一个或者多个连通分量中国至少含有一个点元素为1, 其余为0
	凸壳	$X_k^i = (X_{k-1}^i \odot B) \cup X_{k-1}^i$ 其中 $X_0^i = A$, 当 $X_k^i = X_{k-1}^i$ 时, 过程收敛, 记 $D^i = X_k^i$, $C(A) = \bigcup_{i=1}^s D^i$	寻找图像I中物体A的凸壳C(A)
	细化	$A \otimes B = A - (A \ominus B) = A \cap (A \oplus B)^c$ $A \otimes \{B\} = ((\dots ((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$ $B = \{B^1, B^2, B^3, \dots, B^n\}$	细化集合A,实际工作中常用一系列结构元进行细化
	粗化	$A \odot B = A \cup (A \ominus B)$ $A \odot \{B\} = ((\dots ((A \odot B^1) \odot B^2) \dots) \odot B^n)$ $B = \{B^1, B^2, B^3, \dots, B^n\}$	实际中, 粗化常通过细化算法求补集实现, 即为了对集合A进行粗化, 然后对C进行细化完成后对结果取补集
	骨架	骨架化: $S(A) = \bigcup_{k=0}^K S_k(A)$, 其中: $S_k(A) = (A \ominus kB) - (A \ominus (k+1)B) \odot B$ A的重建: $A = \bigcup_{k=0}^K (S_k(A) \oplus kB)$, 其中: $(S_k(A) \oplus kB)$ 表示从 $S_k(A)$ 开始的k次连续膨胀	A的骨架可以通过腐蚀和开操作来获取, A可以通过骨架子集来重建:
	裁减	$X_1 = A \otimes \{B\}$ $X_2 = \bigcup_{k=1}^s (X_1 \odot B^k)$ $X_3 = (X_2 \oplus H) \cap A$ $X_4 = X_1 \cup X_3$	裁减本质上是对细化和骨架化算法的补充, 因为这些过程往往会留下需要由后处理来清除的某些寄生成分。解决方案是通过不断消除寄生分支的端点来抑制寄生分支