**摘要**

随着人们的生活水平逐步上升，健康问题越来越受到人们的重视。在科学高度发展的今天，防癌治癌成为人们关注的热点话题。目前，结直肠癌的发病率及死亡率有明显的升高趋势，防治结直肠癌成为当代人们关切的问题。本文在基于人体生理学的背景角度出发，使用数学思想及计算方法，借助可实现数据挖掘软件及模型，构建能够在一定程度判断直肠癌、预测病人生存时长的数学模型。对此，我们展开了以下工作：

首先，对于直肠癌影像诊断过程，我们将本题拆解成三步：读取数据并筛选、分割图像；建立模型预测病人的生存时长；对分割结果进行检验。

针对问题一，我们依据题目提供文献，综合赛制期间收集的文献及材料发现，卷积神经网络（CNN）可在读取数据的基础上同时实现图像特征的筛选和图像的分割。而除卷积神经网络外，还可实现该目标功效的有随机森林（Random Forest）。我们考虑到现有的软件及知识储备情况，选择采用matlab软件及其神经网络工具箱实现随机森林模型（Random Forest）的构建，实现病人CT影像中的直肠癌筛选与图像分割。

针对问题二，我们依据题目提供文献了解到对于病人的生存时长可通过logistic 回归、支持向量机（SVM）、生存预测函数（Cox）对病人的生存时长进行测算。

针对问题三，我们借助DICE系数对于我们预测的CT图像上的直肠癌状况与已知的掩盖图像进行比对，借助Python实现预测结果与实际情况的比对，验证该模型对于临床的价值。

最后，我们回顾上述过程，分析模型构建过程中的优缺点，对模型中可能出现的误差进行了原因分析于阐释，反思模型创造的社会意义。此外，我们对于模型提出研究建议与设计改进建议，为发现更好的模型优化方案贡献绵薄之力。

**关键词：**直肠癌；CT影响识别；随机森林模型（Random Forest）；DICOM处理；DICE系数

**英文摘要(稍后写)**

随着人们的生活水平逐步上升，健康问题越来越受到人们的重视。在科学高度发展的今天，防癌治癌成为人们关注的热点话题。目前，结直肠癌的发病率及死亡率有明显的升高趋势，防治结直肠癌成为当代人们关切的问题。本文在基于人体生理学的背景角度出发，使用数学思想及计算方法，借助可实现数据挖掘软件及模型，构建能够在一定程度判断直肠癌、预测病人生存时长的数学模型。对此，我们展开了以下工作：

首先，对于直肠癌影像诊断过程，我们将本题拆解成三步：读取数据并筛选、分割图像；建立模型预测病人的生存时长；对分割结果进行检验。

针对问题一，我们依据题目提供文献，综合赛制期间收集的文献及材料发现，卷积神经网络（CNN）可在读取数据的基础上同时实现图像特征的筛选和图像的分割。而除卷积神经网络外，还可实现该目标功效的有随机森林（Random Forest）。我们考虑到现有的软件及知识储备情况，选择采用matlab软件及其神经网络工具箱实现随机森林模型（Random Forest）的构建，实现病人CT影像中的直肠癌筛选与图像分割。

针对问题二，我们依据题目提供文献了解到对于病人的生存时长可通过logistic 回归、支持向量机（SVM）、生存预测函数（Cox）对病人的生存时长进行测算。

针对问题三，我们借助DICE系数对于我们预测的CT图像上的直肠癌状况与已知的掩盖图像进行比对，借助Python实现预测结果与实际情况的比对，验证该模型对于临床的价值。

最后，我们回顾上述过程，分析模型构建过程中的优缺点，对模型中可能出现的误差进行了原因分析于阐释，反思模型创造的社会意义。此外，我们对于模型提出研究建议与设计改进建议，为发现更好的模型优化方案贡献绵薄之力。

**关键词：**直肠癌；CT影响识别；随机森林模型（Random Forest）；DICOM处理；DICE系数

# 一、问题重述

## 1.1背景

## 1.2问题

# 二、问题分析

## 2.1 问题一：读取数据并筛选、分割图像，进行特征提取

## 2.2问题二：建立模型预测病人的生存时长

## 2.3 问题三：对分割结果进行检验

# 三、基本假设与符号说明

## 3.1基本假设

## 3.2符号说明

# 四、模型的建立与求解

## 4.1模型准备

### **4.1.1关于DICOM格式的图像**：

因为不同物质吸收X射线能力不同，所以便有了CT图像，它根据X光射线来吸收剖面的重构图像，X射线可用于诊断。CT图像被用来做大量的医学影像分析，可供医生窥视到人体内的结构，判断是否出现症状，便于早期诊断筛查。CT图像特异性及敏感性相较于MR图像，要差得多。

CT成像中的伪影包括：部分容积效应、条形伪影、运动伪影、束硬化伪影、环状伪影、金属伪影等。由于这些伪影的存在给CT图像分割带来了一定的难度，不同组织部位分割精度也不一样。

为了研究CT图像中的伪影及还原患者相关数据，便出现了DICOM通讯协议。

CT扫描会测量放射性密度，存在名叫“胡斯菲尔德单元”（HU）的测量的单元。以下是关于CT扫描测量的详细内容：

在voxel内通过计算所有衰减值可以得到平均值，该CT图像中每个像素点都会被分配到这个平均值。而“胡斯菲尔德单元”（HU）是以任意单元的规模显示的，也就是这个与水的衰减值相比较得出的平均值。

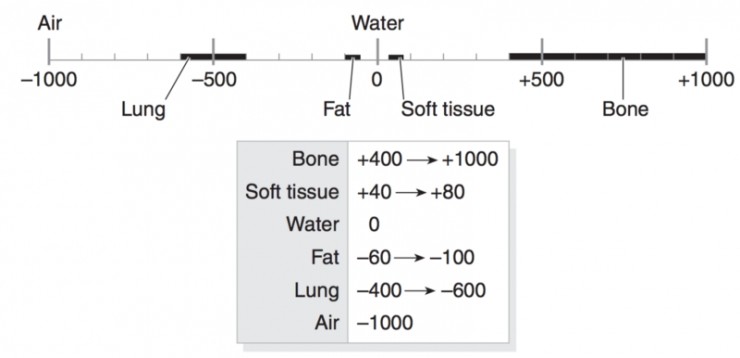
水的衰减值被这个任意单元的规模定为零。CT数字的范围平均是2000HU。每个数值都代表了一种灰色阴影，在光谱两端有+1000白色和-1000黑色。有些扫描仪是有柱状扫描边界的，但是其输出的图像确实方形的。在扫描边界之外的像素将被赋予-2000的定值。

图1. 胡斯菲尔德单元说明

医用图像因为缺乏非常复杂单纯的线性特征，医用图像的自动分隔是很困难的工作，并且，受到音量效果，灰度的不均匀性，艺术幻影，软组织间的灰度及近邻因素等的影响。

通常，图像中单个组织的渐变逐渐变化，并且其概率密度受特定分布函数的影响。 组织的相应图像区域包含有限数量的像素并且满足部分体积平均值。

然而，区域内的单一像素的灰度等级，与哪个类别的灰度等级不同，通常被认为是混在一起的组织类。

当前，一般用于CT和MR图像的分段方法包括基于阈值的分段、基于区域的分段、基于变形模型的分段、模糊分段和基于神经网络的分割。

### **4.1.2**关于图像的分割

如果网络的隐藏层发展到多个层次，则称为深度学习，但同时需要逐层初始化技术来解决深层网络训练的问题。 卷积神经网络（CNN）是由深度学习和图像处理技术相结合创建的经典模型。 实现该模型的网络的示例在处理某些图像问题方面是有效的。

分析医学CT图像时，首先我们要确定直肠癌的特征，确定特征划分标准及提取特征算法；随后，我们对于特征变化进行预测，并检验特征变化预测值；最后我们对于模型弊端进行改良。

在开始预测直肠癌的特征变化时，我们首先将CT图像数据导入matlab中，准备图像成像和分割及特征提取。基于图像数据导入matlab后，我们借助matlab对图像进行了成像与分割。在进行图像成像时，我们先对图像进行灰度归一化处理以加快神经网络的收敛性和求解问题的速度，即使像素的灰度值分布在0到255之间，图像对比度也不够（图像像素亮度分布不平衡），这可能会影响后续处理。

我们基于图像灰度线性归一化对图像实现归一化：

我们借助matlab，直接调用matlab函数实现图像灰度归一化。

*% Another way to normalazation, which only calls MATLAB function*

img3 = mat2gray(oriImage);

figure;

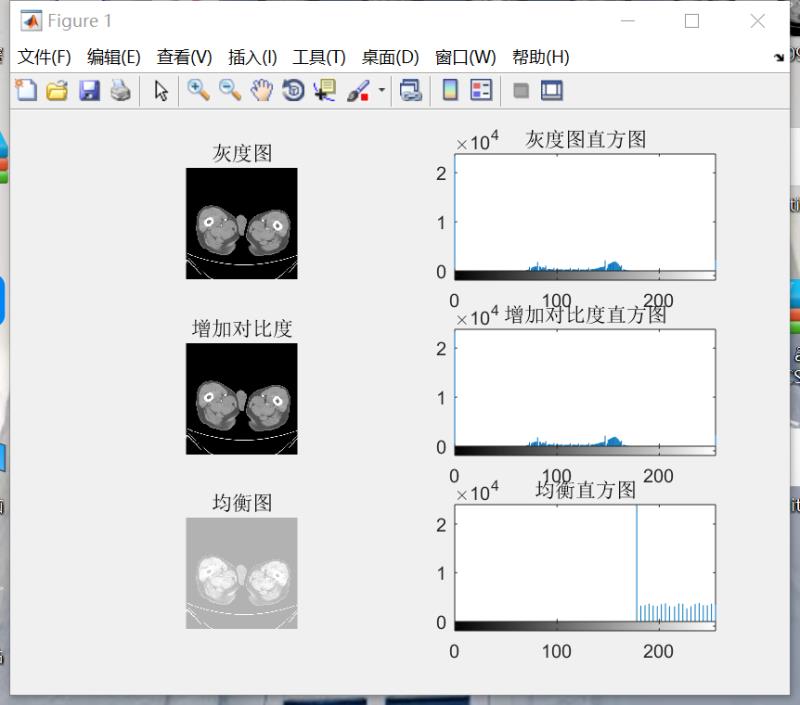
imshow(img3);

4.2求解

4.2.1 问题一的求解

在实现图像的归一化处理后，我们开始进行图像分割和图像特征提取。在图像特征方面，我们关注图像表现的强度、形状、纹理、小波特征。在进行图像特征提取时，我们根据数学方法定义图像特征，然后对其进行计算，并将其放入分类器进行分类。

首先，我们采用直方图将一幅图内的各像素分布情况描绘出来，基于

横坐标为像素，纵坐标为相应的频率，灰度图像的直方图将利用matlab程序得到。

hist = zeros(img\_value,1);

for value = 1:img\_value

hist(value) = length(find((img\_vol\_subvol == value) & (mask\_vol\_subvol == 1) ));

end

*% Compute the relative probabilities from the histogram:*

prob = hist / num\_ROI\_voxels;

*% Compute the histogram\_based metrics:*

texture\_metrics(1:6) = compute\_histogram\_metrics(prob,img\_value);

随后，我们基于直方图的PET/CT纹理特征，进行特征提取。在特征值提取过程中，我们对6个特征代表值进行测算，这6个特征代表值分别有：均值、方差、偏态、峰态、能量、熵，特征获取将用到matlab代码实现。

val\_indices = (1: img\_value)';

indice = find(prob);

vector(1) = sum(val\_indices .\* prob);

vector(2) = sum( ((val\_indices - vector(1)).^2) .\* prob );

if vector(2) > 0

vector(3) = sum( ((val\_indices - vector(1)).^3) .\* prob ) / (vector(2)^(3/2));

vector(4) = sum( ((val\_indices - vector(1)).^4) .\* prob ) / (vector(2)^2);

vector(4) = vector(4) - 3;

else

vector(3) = 0;

vector(4) = 0;

end

vector(5) = sum( prob .^2 );

vector(6) = -sum( prob(indice) .\* log(prob(indice)) );

我们提取到直肠癌的特征后，进行边缘及轮廓计算。我们调用matlab中的sobel函数，求取边缘的同时求取梯度方向特征，

void sobel\_te(IplImage \*img,IplImage \* sobel8u){

IplImage\* sobel=cvCreateImage( cvGetSize(img), IPL\_DEPTH\_16S, 1 );

IplImage\* sobel\_y=cvCreateImage( cvGetSize(img), IPL\_DEPTH\_16S, 1 );

IplImage\* sobel\_x=cvCreateImage( cvGetSize(img), IPL\_DEPTH\_16S, 1 );

cvSobel(img,sobel\_y,1,0,3); //提取Y方向

cvSobel(img,sobel\_x,1,0,3); //提取X方向

cvOr(sobel\_x,sobel\_y,sobel,NULL);

cvConvertScaleAbs(sobel,sobel8u,1,0);//把sobel计算后的值由16位转为8位供正常显示

sobel8u->origin = img->origin;

}

至此，我们得到了直肠癌CT图像的边缘与轮廓特征，实现直肠癌的筛选和提取。

4.2.2 问题二的求解

在对于病人的生存时长方面，我们采用随机森林模型进行预测。

首先，我们将已知的每张病人CT图像作为训练集，每次随机并有序放回训练集中的N个样本，实现迭代抽样，作为树的训练集，并开始对于行和列进行采样。

clear all;

right\_node=cell(3,1);*%3\*1的单元数组*

*% rchild\_value=cell(3,1);%3\*1的单元数组*

*% rchild\_node\_num=cell(3,1);%3\*1的单元数组*

sn=300; *%随机可重复的抽取sn个样本*

tn=10; *%森林中决策树的数目*

load('abc.mat');

n = size(r,1);

*%% 样本训练采用随机森林和ID3算法构建决策森林*

arr = [];

for j=1:tn

sample=randi([1,n],1,sn);*%从1至1000内随机抽取sn个样本*

SData=r(sample,:);

[tree,arr]= train\_C4\_5(SData, 5, 10, arr);

right\_node{j,1}=tree;

end

load('abc.mat');

T = r;

TData = roundn(T,-1);

*%统计函数，对输入的测试向量进行投票，然后统计出选票最高的标签类型输出*

result = statistics(tn, right\_node, TData, arr);

gd = T(:,end);

len = length(gd);

count = sum(result==gd);

fprintf('共有%d个样本','判断正确的有%d\n',len,count);

之后，我们对于训练集中的树进行迭代抽样，展开训练。

我们需要设置五个变量

train\_ptn - Train patterns 训练样本 每一列代表一个样本 每一行代表一个特征

train\_trg - Train targets 1×训练样本个数 每个训练样本对应的判别值

test\_patterns - Test patterns 测试样本，每一列代表一个样本

node - Percentage of incorrectly assigned samples at a node 一个节点上未正确分配的样本的百分比经过行采样和列采样之后，我们对于随机森林模型进行测试。

train\_ptn = S(:,1:end-1)';

train\_trg = S(:,end)';

[Ni, M] = size(train\_ptn); *%M是训练样本数，Ni是训练样本维数，即是特征数目*

node = node\*M/100; *% 5\*训练样本数目/100*

Nu --用于确定变量是离散还是连续（该值始终设置为10）

这里用10作为一个阈值，如果某个特征的无重复的特征值的数目比这个阈值还小，就认为这个特征是离散的

之后需要查找哪些输入模式(特征)是离散的，并离散测试模式上的相应维度

定义一个变量Nb，通过遍历遍历每个特征得到无重复的特征值的数目，如果这个特征的无重复的特征值的数目比这个阈值还小，就认为这个特征是离散的

之后定义递归构造树，将特征值存入，进行训练样本对应的判别标签，为防止过拟合，将node值设置为5-10，表示样本数小于一定阈值结束递归

T = make\_tree(train\_ptn, train\_trg, node, dim, max(dim), 0, flag);

function T = make\_tree(patterns, targets, node, dim, maxNbin, base, flag)

然后计算当前的信息熵

Inode = -sum(Pnode.\*log(Pnode)/log(2));

再次遍历特征值，得到当前所有样本的这个特征的特征值、无重复的特征值构成向量Ud及得到无重复的特征值的数目。

如果这个特征是离散特征，那么计算信息增益率(GainRatio)，公式为Gain(A)/I(A)。

其中，Gain(A)=Inode-sum(Pk.\*info)就是属性A的信息增益，

I(A)=-sum(Pk.\*log(eps+Pk)/log(2))为属性A的所包含的信息。

P = zeros(length(Uc), Nbins); *%Uc是判别标签的个数 判别标签个数×无重复的特征值的数目*

for j = 1:length(Uc) *%遍历每个标签*

for k = 1:Nbins *%遍历每个特征值*

indices = find((targets == Uc(j)) & (patterns(i,:) == Ud(k)));

*% &适用于矩阵间的逻辑运算 &&不适用，只能用于单个元素 &的意思也是与*

*%找到 （样本标签==第j个标签 并且 当前所有样本的这个特征==第k个特征值） 的样本个数*

P(j,k) = length(indices); *%记为P(j,k)*

end

end

Pk = sum(P); *%取P的每一列的和，也就是得到当前所有样本中，这个特征的特征值==这个特征值的样本数目 Pk(1×特征值数目)表示这个特征的特征值等于每个特征值的样本数目*

P1 = repmat(Pk, length(Uc), 1); *%把Pk复制成 判别标签个数 行*

P1 = P1 + eps\*(P1==0); *%这主要在保证P1作被除数时不等于0*

P = P./P1; *%得到当前所有样本中，这个特征的值等于每个特征值且标签等于每个标签的样本，占当前这个特征值中的样本的比例*

Pk = Pk/sum(Pk); *%得到当前所有样本中，这个特征的值等于每个特征值的样本，占当前样本总数的比例*

info = sum(-P.\*log(eps+P)/log(2)); *%对特征集中每个特征分别计算信息熵 info(1×特征值数目)*

delta\_Ib(i\_idx) = (Inode-sum(Pk.\*info))/(-sum(Pk.\*log(eps+Pk)/log(2))); *%计算得到当前特征的信息增益率*

对于连续特征(主要要找到合适的分裂值，使数据离散化)，那么就计算分裂信息度量

P = zeros(length(Uc), 2); *% P(判别标签数目×2) 列1代表前..个样本中的标签分布情况 列2代表除前..个样本之外的标签分布情况 这个..就是各个分裂位置*

*%Sort the patterns*

[sorted\_data, indices] = sort(data); *%data里存放的是当前所有训练样本的这个特征的特征值 从小到大排序 sorted\_data是排序好的数据 indices是索引*

sorted\_targets = targets(indices); *%当然，判别标签也要随着样本顺序调整而调整*

*%Calculate the information for each possible split 计算分裂信息度量*

I = zeros(1,Nbins);

delta\_Ib\_inter = zeros(1, Nbins);

for j = 1:Nbins-1

P(:, 1) = hist(sorted\_targets(find(sorted\_data <= Ud(j))) , Uc); *%记录<=当前特征值的样本的标签的分布情况*

P(:, 2) = hist(sorted\_targets(find(sorted\_data > Ud(j))) , Uc); *%记录>当前特征值的样本的标签的分布情况*

Ps = sum(P)/L; *%sum(P)是得到分裂位置前面和后面各有样本数占当前样本总数的比例*

P = P/L; *%占样本总数的比例*

Pk = sum(P); *%%sum(P)是得到分裂位置前面和后面各有多少个样本 比例*

P1 = repmat(Pk, length(Uc), 1); *%把Pk复制成 判别标签个数 行*

P1 = P1 + eps\*(P1==0);

info = sum(-P./P1.\*log(eps+P./P1)/log(2)); *%计算信息熵（分类期望信息）*

I(j) = Inode - sum(info.\*Ps); *%Inode-sum(info.\*Ps)就是以第j个样本分裂的的信息增益*

delta\_Ib\_inter(j) = I(j)/(-sum(Ps.\*log(eps+Ps)/log(2))); *%计算得到当前特征值的信息增益率*

end

之后，找到当前要作为分裂特征的特征，找到所有特征中最大的信息增益对应的特征，然后得到选择的这个作为分裂特征的特征的那一行 也就是得到当前所有样本的这个特征的特征值以及得到这个特征的无重复的特征值的数目

详细代码代码说明解释见附录

function [result] = statistics(tn, rnode, PValue, discrete\_dim)

TypeArr = {'1','2'};

TypeNum = [0 0];

test\_patterns = PValue(:,1:end-1)';

num = length(TypeNum);

type = zeros(tn,size(test\_patterns,2));

for i = 1:tn

type(tn,:) = vote\_C4\_5(test\_patterns, 1:size(test\_patterns,2), rnode{i,1}, discrete\_dim, num);

end

result = mode(type,1)';

end

最后，我们通过随机森林模型判定的直肠癌特征情况，预测病人的生存时长，最终得到样本测试结果。

function targets = vote\_C4\_5(patterns, indices, tree, discrete\_dim, Uc)

trgs = zeros(1, size(patterns,2)); *%设置每个样本的初始预测标签都是0*

if (tree.dim == 0) *%这说明达到了树的叶子节点*

*%Reached the end of the tree*

trgs(indices) = tree.child; *%得到样本对应的标签是tree.child*

return

end

dim = tree.dim; *%得到分裂特征*

dims= 1:size(patterns,1); *%得到特征索引*

if (discrete\_dim(dim) == 0) *%如果当前分裂特征是个连续特征*

*%Continuous pattern*

in = indices(find(patterns(dim, indices) <= tree.split\_loc)); *%找到当前测试样本中这个特征的特征值<=分裂值的样本索引*

trgs = trgs + vote\_C4\_5(patterns(dims, :), in, tree.child(1), discrete\_dim(dims), Uc); *%对这部分样本再分叉*

in = indices(find(patterns(dim, indices) > tree.split\_loc)); *%找到当前测试样本中这个特征的特征值>分裂值的样本索引*

trgs = trgs + vote\_C4\_5(patterns(dims, :), in, tree.child(2), discrete\_dim(dims), Uc); *%对这部分样本再分叉*

else *%如果当前分裂特征是个离散特征*

Uf = unique(patterns(dim,:)); *%得到这个样本集中这个特征的无重复特征值*

for i = 1:length(Uf) *%遍历每个特征值*

if any(Uf(i) == tree.Nf) *%tree.Nf为树的分类特征向量 当前所有样本的这个特征的特征值*

in = indices(find(patterns(dim, indices) == Uf(i))); *%找到当前测试样本中这个特征的特征值==分裂值的样本索引*

trgs = trgs + vote\_C4\_5(patterns(dims, :), in, tree.child(find(Uf(i)==tree.Nf)), discrete\_dim(dims), Uc);*%对这部分样本再分叉*

end

end

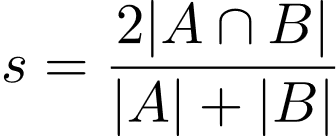
end

4.2.3 问题三的求解

关于图像的分析：

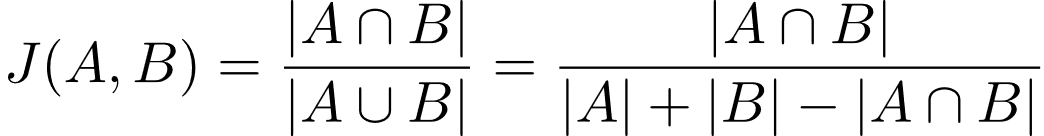
我们可以把字符串理解为一种集合，Dice距离用于度量两个集合的相似性，故也能用于度量字符串的相似性。

Dice系数定义如下：

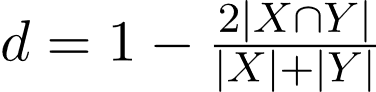


将A和B的交集数的两倍作为分子，x和y的长度之和作为分母，所以范围在0和1之间。

从公式上来看，Jaccard和Dice系数是类似的。Jaccard是在分子和分母上都减去了|A∩B|。

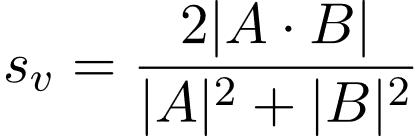


与Jaccard不同的是，相应的差异函数



不是一个合适的距离度量措施，因为它没有三角形不等性的性质。例如给定 {a}, {b}, 和 {a,b}, 前两个集合的距离为1， 而第三个集合和其他任意两个集合的距离为三分之一。

与Jaccard类似, 集合操作可以用两个向量A和B的操作来表示:



利用Python，实现代码为：

def dice\_coefficient(a, b):

"""dice coefficient 2nt/na + nb."""

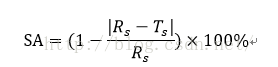
a\_bigrams = set(a)

b\_bigrams = set(b)

overlap = len(a\_bigrams & b\_bigrams)

return overlap \* 2.0/(len(a\_bigrams) + len(b\_bigrams))

关于分割精度：



其中参考面积为Rs，它是被手工勾画出的分割图像，图像的真实面积通过算法分割来得到，为Ts，错误分割的像素点个数为|Rs-Ts|。

存在分割率，其定义为分割在图像参考面积之外的像素点的比率：

https://img-blog.csdn.net/20161101002459046?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

该含义与上述相同，OS表示不应包括在分割结果中的像素数，实际上是分割结果的像素数。 也就是说，OS中的像素点实际上出现在分割图像中，并且理论上不分割图像RS。

同时也存在欠分割率，其定义为分割在GT图像参考面积之中欠缺的像素点的比率：https://img-blog.csdn.net/20161101002503838?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

意思和上面一样。Us表示分段结果中必须包含的像素数，但实际上表示分段结果中不包含的像素数。即，Us的像素在理论上的分段图像中被显示,不过，在实际的分段图像中不显示。

5.模型评价与改进

5.1 误差分析

5.2 模型评价

5.3 模型改进

6.参考文献

[1]李铭.CT金属伪影校正研究[D].长春:中国科学院研究生院(长春光学精密机械与物理研究所),2015.

[2]于海峰.医学显微图像识别与分析[J].计算机光盘软件与应用,2011,1:73-75.

[3]王洁.基于图学习的医学图像标注方法[J].中国电子商情：通信市场,2013.

[4]赵志平,陈雷月.基于深度传感器图像分割技术的研究[J].信息技术,2014,1:109-112.

[5]张海涛.新的时间序列相似性度量方法[J].计算机工程与设计,2014,1:63-66.

[6]李扬定.基于典型机器学习模型的图像检索算法[D].广西:广西师范大学,2013.