



上海交通大学硕士学位论文

# 肺部 CT 平扫图像中的支气管分割技术研究

姓 名：徐 赞

学 号：116033210067

导 师：卢宏涛

企业导师：刘维平

院 系：电子信息与电气工程学院

学 科/专 业：计算机科学与工程

申 请 学 位：工程硕士

2023 年 5 月 3 日



**A Dissertation Submitted to  
Shanghai Jiao Tong University for Master Degree**

**RESEARCH ON THE TECHNIQUE OF PULMONARY  
BRONCHUS SEGMENTATION IN THE  
CT-SCANNING IMAGES**

**Author:** Xu Zan

**Supervisor:** Lu Hong Tao

**Enterprise Supervisor:** Liu Wei Ping

School of Electronic Information and Electric Engineering

Shanghai Jiao Tong University

Shanghai, P.R. China

May 3<sup>rd</sup>, 2023



## 上海交通大学

### 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全知晓本声明的法律后果由本人承担。

学位论文作者签名：

日期： 年 月 日

## 上海交通大学

### 学位论文使用授权书

本人同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。

本学位论文属于：

- 公开论文**
  - 内部论文**，保密  1年/ 2年/ 3年，过保密期后适用本授权书。
  - 秘密论文**，保密 \_\_\_\_ 年（不超过 10 年），过保密期后适用本授权书。
  - 机密论文**，保密 \_\_\_\_ 年（不超过 20 年），过保密期后适用本授权书。
- (请在以上方框内选择打“√”)

学位论文作者签名：

日期： 年 月 日

指导教师签名：

日期： 年 月 日



## 摘要

肺部疾病诊断与筛查通常使用计算机断层 CT 扫描患者的肺部区域，获得病灶部位的切片图像，临床医生根据 CT 扫描图像诊断疾病。本文的研究对象是患者肺部的 CT 扫描图像，目标是分割提取出精确的支气管气道树的三维模型。支气管气道树是很多肺病诊疗的重要参考，目前基于卷积神经网络的图像分割技术在气道树分割提取方面面临严重类别不平衡等挑战。本文提出两种新方法，注意力蒸馏方法和通道级特征再学习方法，来解决这些问题挑战。本文构建以 3D-UNet 网络为基础的基准模型，采用 ATM22 数据集在 AI 超算平台上训练分割模型。

本文在以 3D-UNet 为基准网络对支气管气道树进行分割的实验中发现存在假阳性误检和假阴性漏检的不足，因支气管体素标注数据极度稀疏而致。本文受注意力迁移和知识蒸馏启发而提出的注意力蒸馏方法帮助模型聚焦于支气管树状结构特征，摒弃大量无用背景的干扰，提取出气道树枝干和细小分支。本文在 3D-UNet 上采样路径增加注意力蒸馏模块，改善 3D-UNet 网络对特征的聚焦问题，实验结果表明注意力蒸馏方法可有效降低假阳性率和假阴性率。

支气管气道树中同时存在枝干支气管体素的粗糙特征和末梢支气管体素的精细特征，精细特征会因池化层降低分辨率而逐渐被“擦除掉”。不同空间位置有不同的重要性，本文提出通道级特征再学习方法通过把卷积层输出的特征通道重新组合，在训练过程中使重要的气道特征增加权重而被重视，使无关的区域降低权重而被忽略。本文在 3D-UNet 网络的每个卷积层和池化层之间插入特征再学习模块，改进对精细特征的识别问题。综合实验结果表明末梢支气管分割良好，性能指标得到明显提升。

**关键词：**支气管气道树分割，3D-UNet 基准网络，注意力蒸馏，通道级特征再学习，三维卷积层



## ABSTRACT

Diagnosing and screening the pulmonary diseases usually utilizes the computed tomography scanning on lung region, captures the image slices of lesion part, then clinicians make the diagnosis according to these CT-scanning images. In this thesis we investigate the pulmonary CT-scanning images, for the goal of extracting the accurate airway tree 3D model. Bronchial airway tree is an important reference for the treatment of many pulmonary diseases. Currently CNN-based airway tree segmentation is facing some challenges like severe class imbalance. We have proposed 2 new methods: attention distillation (AD) method and channel-wise feature re-learning (FRL) method, in this thesis to tackle the problem. Based on the 3D-UNet baseline network, we adopt the ATM22 dataset to train a segmentation model on the AI supercomputer platform.

An experiment based on 3D-UNet baseline network, which we have done to segment the bronchial airway, had shown the deficiencies of false positive and false negative detecting. It is because of the extremely sparse bronchial voxels annotated on CT volumetric data. Motivated by the attention transfer and knowledge distillation, the attention distillation method is proposed to focus on the tree structure of bronchial airway. It can discard the interference of massive useless background, extract the main airway branches and some small ones. We add AD modules onto the decoder path of 3D-UNet baseline network, in order to improve the focusing issue on important airway features. Some experiments had shown that attention distillation method can effectively reduce the FNR and FPR.

The airway tree has both the coarse features of voxels on trunk bronchus and the fine features of voxels on distal bronchus, the fine features could be gradually erased due to resolution reduction in the pooling layers. Different spatial positions have different degrees of importance. FRL method is proposed to recombine the feature channels which were produced from convolutional layer, it has the crucial airway regions preferred gradually with higher weights, meanwhile has the uninformative regions negleted with lower weights. The FRL module is inserted between each convolutional layer and pooling layer of 3D-UNet baseline network, just to improve the ability of recognizing more refined features, after that a comprehensive experiment showed that the distal bronchus had been well segmented, and the performance metrics were improved obviously.

**Key words:** Bronchial airway segmentation, 3D-Unet baseline network, Attention distillation, Channel-wise feature re-learning, 3D convolutional layer

# 目 录

<b>第一章 绪论 .....</b>	<b>1</b>
1.1 研究背景和意义 .....	1
1.2 本研究的应用前景 .....	3
1.3 研究现状和发展趋势 .....	4
1.3.1 传统的图像分割方法 .....	4
1.3.2 基于深度学习的图像分割方法 .....	6
1.3.3 针对肺部支气管气道树的医疗图像分割方法 .....	10
1.4 挑战和困难 .....	13
1.5 本文的研究内容和创新点 .....	14
1.6 论文组织结构 .....	15
<b>第二章 基准分割模型 3D-UNet 和分割效果分析 .....</b>	<b>17</b>
2.1 医疗图像语义分割网络层次 .....	17
2.1.1 卷积层 .....	18
2.1.2 归一化层 .....	18
2.1.3 池化层 .....	20
2.2 3D-UNet 基准模型 .....	22
2.2.1 网络结构设计 .....	22
2.2.2 损失函数、优化器与学习率调整 .....	24
2.2.3 模型实现和运行环境 .....	30
2.3 数据集 ATM22 .....	31
2.4 评价指标和分割效果可视化 .....	33
2.5 实验结果与分析 .....	37
2.5.1 3D-UNet 基准网络训练过程 .....	37
2.5.2 实验结果和分析 .....	40
2.6 本章小结 .....	45
<b>第三章 注意力蒸馏方法对基准模型的改进 .....</b>	<b>47</b>
3.1 注意力蒸馏方法 .....	48

---

3.2 注意力蒸馏的效果可视化.....	50
3.3 引入注意力蒸馏模块对 3D-UNet 基准网络进行改进 .....	51
3.4 对比实验与实验结果分析.....	55
3.5 本章小结 .....	58
<b>第四章 通道级特征再学习方法提取精细支气管特征 .....</b>	<b>61</b>
4.1 通道级特征再学习方法 .....	62
4.2 引入通道级特征再学习模块对 3D-UNet 基准网络进行改进 .....	66
4.3 综合实验 .....	68
4.4 性能指标对比 .....	69
4.5 本章小结 .....	71
<b>第五章 总结与展望 .....</b>	<b>73</b>
5.1 全文工作总结 .....	73
5.2 研究展望 .....	74
<b>参考文献 .....</b>	<b>77</b>
<b>附录 A 实验的部分源代码.....</b>	<b>83</b>
<b>致 谢 .....</b>	<b>87</b>
<b>学术论文和科研成果目录 .....</b>	<b>89</b>

## 插 图

图 1-1 肺部支气管气道树的分支结构 .....	2
图 1-2 CT 扫描仪坐标系, 人体坐标系和 DICOM 图像坐标系对应关系 .....	3
图 1-3 ION 肺部导管机器人 .....	4
图 1-4 医疗图像 (DICOM 格式的灰度图) .....	5
图 1-5 FCN 网络的结构 .....	6
图 1-6 卷积下采样路径 vs. 反卷积上采样路径 .....	7
图 1-7 卷积与反卷积, 池化与反池化的操作 .....	7
图 1-8 经典的图像分割 U-Net 网络结构 .....	8
图 1-9 一种通用的轻量化的 UNet 网络 .....	8
图 1-10 稠密型注意力机制网络用来检测 COVID-19 的感染情况 .....	9
图 1-11 对比正常肺部、普通肺炎和 COVID-19 新冠肺炎的阴影, 突出新冠肺炎感染区域 .....	10
图 1-12 UNET 网络基本结构 .....	10
图 1-13 TACNet 气道树分割网络 .....	11
图 1-14 多信息融合 Mif-CNN 网络和基于 CNN 区域生长网络的组合 .....	12
图 1-15 创新性的树网络结构 .....	12
图 2-1 卷积神经网络基本构成 .....	17
图 2-2 三维卷积操作的原理 .....	19
图 2-3 四种归一化方法的差别 .....	20
图 2-4 池化操作 .....	21
图 2-5 池化层按通道独立计算 .....	21
图 2-6 3D-UNet 下采样路径上卷积层块和池化层的结构 .....	23
图 2-7 3D-UNet 上采样路径上反卷积层块和池化层的结构 .....	24
图 2-8 本文所设计与使用的 3D-UNet 网络结构 .....	25
图 2-9 易于分割的枝干气管和难以分割的末梢支气管 .....	27
图 2-10 AI 超算平台 Slurm 作业调度系统 .....	31
图 2-11 裁切 CT 图像切片的效果对比 .....	32
图 2-12 支气管气道树分割的 3D 模型 .....	36
图 2-13 3D-UNet 基准网络训练时的损失函数曲线 .....	38

---

图 2-14 验证集 9 例 CT 图像的支气管气道树分割可视化 3D 模型 .....	44
图 3-1 末梢支气管出现假阴性分割效果 .....	47
图 3-2 注意力蒸馏的原理 .....	49
图 3-3 注意力蒸馏后的效果 .....	51
图 3-4 注意力蒸馏后图像特征对比 .....	52
图 3-5 3D-UNet + Attention Distillation 网络结构示意图 .....	53
图 3-6 3D-UNet + AD 网络结构支气管气道树分割可视化 3D 模型 .....	56
图 3-7 支气管内外壁标注差异而导致覆盖问题 .....	58
图 3-8 ATM_074_0000 和 ATM_271_0000 两个特殊病例的分割结果分析 .....	59
图 4-1 相同体积的切割体在不同空间位置获得的体素密度差异 .....	61
图 4-2 卷积层块的特征的维数 .....	63
图 4-3 特征立方体的空间积分过程示意图 .....	64
图 4-4 3D-UNet + Attention Distillation + Feature Re-Learning 网络结构示意图 ...	67
图 4-5 综合实验验证集和测试集的气道树分割效果一览 .....	70

**表 格**

表 2-1 支气管体素占比.....	28
表 2-2 动态降低学习率.....	30
表 2-3 人工智能超算平台资源 .....	30
表 2-4 评价指标计算方法说明 .....	35
表 2-5 训练集、验证集和测试集的数据一览.....	37
表 2-6 3D-UNet 基准网络训练、验证、测试耗时一览表 .....	39
表 2-7 验证集 CT 切片图像分割效果与指标横向比较 .....	41
表 2-8 验证集 ATM_054_0000 病例第 264 张切片图像分割效果与指标纵向比较	42
表 2-9 测试集上 3D-UNet 基准网络对支气管气道树分割的性能指标表现 .....	43
表 2-10 验证集 9 例 CT 图像的支气管气道树分割指标 .....	45
表 3-1 3D-UNet + AD 网络结构的验证集指标数据一览表.....	55
表 3-2 3D-UNet + AD 网络结构的测试集指标数据一览表.....	57
表 3-3 3D-UNet 与 3D-UNet + AD 的性能指标对比 .....	57
表 4-1 综合实验证集和测试集的评价指标一览 .....	69
表 4-2 性能指标比较表.....	71



## 算 法

算法 3-1 注意力蒸馏的梯度损失..... 54



# 第一章 绪论

## 1.1 研究背景和意义

医疗图像分割技术是计算机视觉和模式识别技术应用在诸如 CT（计算机断层扫描成像）、MRI（磁共振成像）和 Ultrasound（超声成像）等医疗图像上，针对特定的目标，譬如肺部动脉/静脉血管、乳腺肿块，肺癌结节，进行逐体素 (Voxel) 的分类。本文聚焦肺部 CT 扫描图像，研究肺部支气管树状结构的气道分割提取技术。

肺部和呼吸系统疾病对人体健康威胁巨大，自 2019 年底、2020 年初爆发的全球性新冠肺炎 COVID-19 疫情严重冲击人类的公共卫生与生命健康。三年多来，世界各国在抗击新冠肺炎疫情方面付出了惨重的生命代价和社会经济损失。新型冠状病毒经由空气为媒介传播，通过呼吸道进入肺部并感染细胞。包括肺癌在内的一些肺部疾病，慢性阻塞性肺疾病 (COPD)<sup>[1]</sup>、急性呼吸窘迫综合症<sup>[2]</sup>、闭塞性细支气管炎<sup>[3]</sup>、特发性肺纤维化<sup>[4]</sup>、肺挫伤<sup>[5]</sup>等导致肺支气管气道树的形态学变化。支气管气道树的形态学三维模型通过从基于 CT 扫描的肺部图像中精确分割出来，是分析包括哮喘、支气管扩张和肺气肿在内的肺部疾病的关键步骤。测量精确分割出来的支气管的气道管腔尺寸和管壁厚度，可以用于辅助诊断肺栓塞<sup>[6]</sup>，揭示慢性阻塞性肺部疾病 COPD 患者的异常病症。除了上述这些疾病诊断治疗之外，本文研究肺部支气管气道树分割的一个直接动因是为了开发支气管镜和支气管导航与活体检测的手术机器人。手术机器人驱动细长柔性的支气管镜器械，伸入支气管，沿着气道导航至疑似（肺部）病灶部位，对病灶处细胞进行活体采样返回，进而化验诊断出疾病。**我们正在**开发国产的支气管镜导航/检测的手术机器人，而支气管气道树的精确分割就是非常重要的前置研发步骤。构建出支气管气道树的分割模型后，输入患者的肺部 CT 扫描图像数据，就可以输出包含精确的三维空间信息的支气管气道树结构。将这个支气管气道树的结构模型输入给支气管镜导航机器人 (Bronchoscopy and endobronchical navigation robot)，在医生的监视与操纵下，机器人就可以被引导进入支气管管腔内，进行疾病的诊断与治疗了。

**支气管气道树的分支结构。**本文研究支气管气道树的分割，有必要先了解一下支气管树状的气道解剖结构。从图1-1中可以看出，支气管气道树的分支结构非常复杂，从气管 (Trachea) 往下分为左、右两支主支气管 (Main bronchus)，左右两支主支气管各自分岔为上叶支气管 (Superior lobar bronchus)、中叶支气管 (Middle lobar bronchus)

和下叶支气管 (Inferior lobar bronchus)。在每个肺叶内，起于肺叶的支气管继续分岔为多个肺段支气管 (Segmental bronchi)。肺段支气管分为更细小的小叶支气管 (Lobular bronchi)，最后再次分岔出细支气管 (Bronchiole) 到达肺泡。肺部支气管结构是一个具有 6 级的树状分支结构。

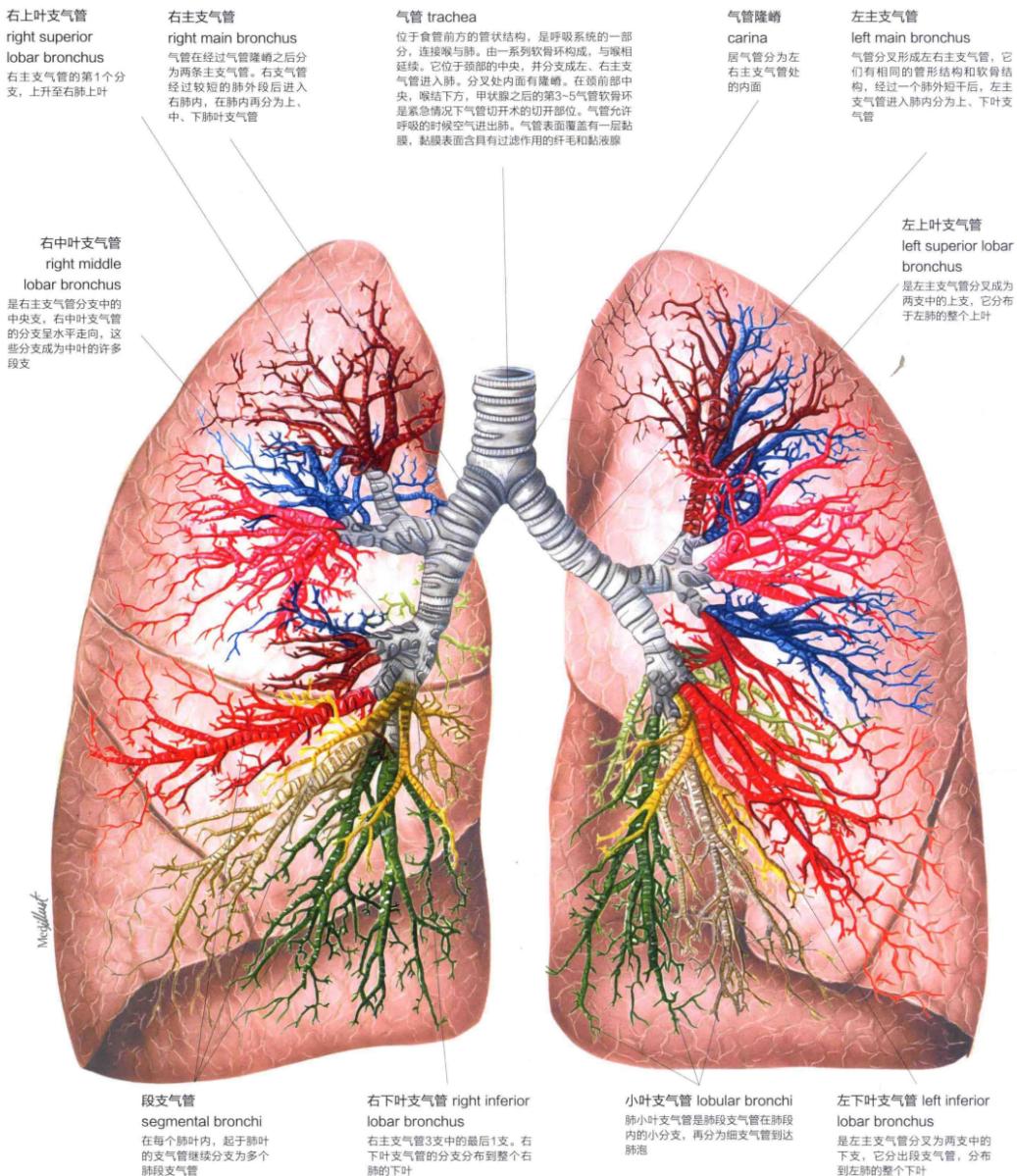


图 1-1 肺部支气管气道树的分支结构<sup>[7]</sup>

Figure 1-1 The branch structure of pulmonary bronchial airway tree

如此繁复，从粗到细不断分化，面对如此稠密细小的支气管气道树，进行分割前需要经验丰富的临床专家进行非常细粒度的标注。手工标注耗费大量时间，成本高昂

且支气管越细小，标注越困难，越是易于出错。标注任务艰巨繁重，急需开发新的算法或模型来帮助临床医生解决支气管气道树分割的问题。

## 1.2 本研究的应用前景

肺部支气管气道树的自动化分割是呼吸系统肺部疾病诊断治疗的一个重要问题和研究领域，在实际应用中具有非常重要的作用，可提高医疗技术水平。当前，在作者所在的医疗机器人行业，本研究项目有如下的应用前景：

- **导管手术机器人** 由于肺部支气管气道树三维模型具有精确的空间位置信息，CT 扫描建立的坐标系赋予 DICOM 医疗图像每个像素都有相应的坐标位置信息，如图1-2所示。这样，分割出来的支气管气道树三维模型其每一个体素 (Voxel) 就

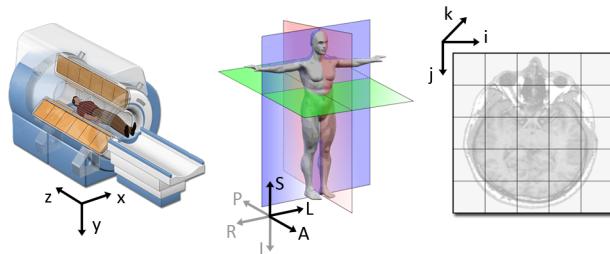


图 1-2 CT 扫描仪坐标系，人体坐标系和 DICOM 图像坐标系对应关系 [8-9]

Figure 1-2 The coordinate system between CT scanner, human body and DICOM imaging

具有精确的坐标位置信息，就可以计算出支气管管腔的中心线位置。

导管机器人就是沿着管腔中心线的路径移动，导航至病灶位置。目前，美国 Intuitive Surgical Company 已经开发出肺部导管机器人 ION，并投入了临床应用。

**我们**正在开发国产的肺部导管机器人，已经完成全国首例机器人辅助经支气管镜肺结节活检手术的**临床实验**。

当然导管手术机器人除了依赖支气管气道树三维模型导航，还需要术中实时定位、支气管镜视觉导航技术的辅助。

- **智慧医疗辅助诊断** 基于 CT 扫描图像的医疗图像分割需要高质量的标注数据，这很大程度上依赖经验丰富的临床医生/专家的专业知识。为了减轻临床医生的标注压力和负担，同时亦为了减少误诊和漏诊的情况发生，医疗图像分割技术在医学辅助诊断上已经获得越来越多的应用。具体到支气管气道树分割技术上，已经被用来辅助诊断一些慢性阻塞性肺部疾病 COPD，支气管扩张和肺气肿等

图 1-3 ION 肺部导管机器人<sup>[10]</sup>

Figure 1-3 ION bronchoscopy robot

一些肺部疾病。随着 AI 图像分割技术的不断进步，可以辅助临床医生更准确更高效地诊断疾病，逐渐达成智慧医疗，提高医疗技术水平和能力。

### 1.3 研究现状和发展趋势

图像分割技术已经发展很多年了，从过去数十年的发展来看，图像分割技术的发展经历了传统的图像分割方法和基于深度学习的分割方法 2 个阶段。传统的图像分割方法产生了一些比较经典的算法，诸如基于阈值的最大类间方差 Otsu 方法<sup>[11]</sup>，基于聚类的 K-Means 算法<sup>[12]</sup>等。而随着近年来深度学习的发展，深度学习应用于图像分割则是涌现了大量优秀的分割方法。Jonathan Long 等人<sup>[13]</sup>首次提出的全卷积网络 (Fully Convolution Neural Network, FCN)，最经典的当属 Olaf Ronneberger 等人<sup>[14]</sup>提出的 UNet 结构，以及在 UNet 基础上改进提出的 V-Net<sup>[15]</sup>结构模型、U-Net++<sup>[16]</sup>结构模型，还有针对医疗影像这类三维体数据的分割模型 3D-UNet 结构<sup>[17]</sup>，诸如这些基于深度学习的图像分割方法，其性能、精度、鲁棒性等指标都在不断进步，发展非常活跃。

下面分别介绍传统的图像分割方法、基于深度学习的图像分割方法和具体针对肺部支气管气道树的分割方法，了解它们的基本思想和发展趋势。

#### 1.3.1 传统的图像分割方法

传统图像分割的方法是根据图像特征设计相应的算法，对每个像素点进行分类。这些图像的特征包括颜色、纹理、亮度和形状等，其本质是根据特定标准确定一个合理的阈值，将每一个像素点与此阈值比较，以确定每一个像素点的分类。综合分析传

统的图像分割方法，我们可以总结出如下的划分：

### 1. 阈值法

阈值法是根据图像的灰度特征信息进行与阈值比较的计算。对欲分割的图像设定一个或多个（不同）的阈值，然后将图像上的每一个像素点与阈值进行比较，得到每个像素点所属的类别。这方面主要的工作有 Sauvola 法<sup>[18]</sup>、最大类间方差 Otsu 法<sup>[11]</sup>等。医疗图像（DICOM 格式<sup>[19]</sup>）都是典型的灰度图（如图1-4所示），阈值法很多都是根据灰度来确定阈值的。

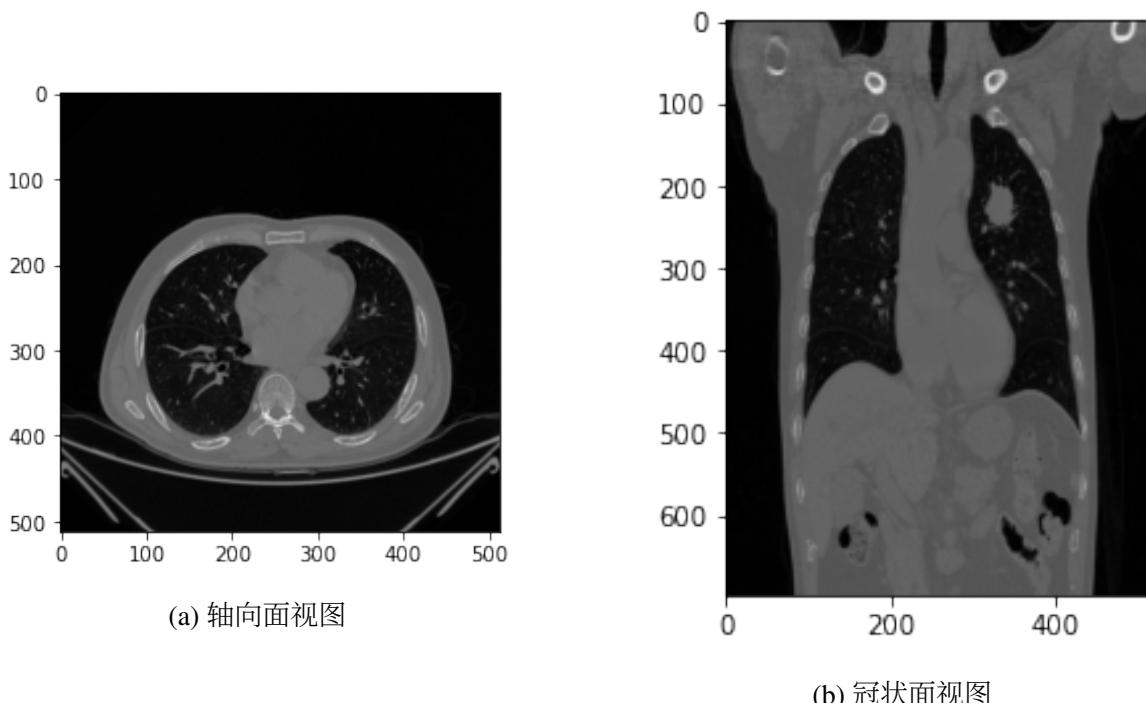


图 1-4 医疗图像 (DICOM 格式的灰度图)

Figure 1-4 Medical image in DICOM format

### 2. 区域法

区域法是根据像素的灰度和纹理信息，以区域一致性规则进行分割。其包括：区域生长法、区域分裂、区域合并法。区域生长法通过指定一个种子点，向生长区域不断地添加满足约束条件的新的像素，直到填满或无法再添加。这些新添加进来的像素即属于一个类别，在不同的生长区域的像素从属于不同的类别。区域法初始形式简单，计算速度也快，它比较适合灰度均匀的图像分割。

### 3. 聚类法

聚类方法则根据像素间的特征相似程度进行迭代式的分类，将特征值相近的一

组像素归为一个类别，然后计算这一组像素的中心位置，通过不断更新迭代直到完成所有像素的分类。这种方法的代表性工作便是 K-Means 算法<sup>[12]</sup>，聚类法使同类像素样本尽可能相近，而异类像素样本则拉大差异。聚类法的实现较为简单，但缺点就是对噪声很敏感，稍微不同的聚类中心和类别选择等因素都会导致分割结果差异，鲁棒性较差。

#### 4. 图论法

图论法是源于聚类的分割方法，它是将图像转换成带权重的无向图，将无向图划分为各种子图，使每个子图内部保持最大相似性，而子图之间则是差异尽可能大，每个子图代表一个类别。相关的图论分割方法就有归一化图分割、最小图分割、迭代式图分割等。

### 1.3.2 基于深度学习的图像分割方法

传统的图像分割方法围绕着图像的某一个具体特征进行精心挑选，设计与之匹配的算法，不具备广泛性和普遍性。对于医疗 CT 扫描图像，如前文图1–4 所述的都是对比度小的灰度图，传统的图像分割方法对于医疗图像不再适合。

近年来随着深度学习技术在计算机视觉领域的应用，飞速发展并取得显著的进步。基于深度卷积神经网络的语义分割方法对图像中的所有像素进行精确预测，预测像素属于哪一个类别的概率，赋予其标签，通过不断地迭代，从而分割出来图像。自美国加州大学 UC Berkeley 的 Jonathan Long 等人<sup>[13]</sup>首次提出端到端、像素到像素的全卷积网络（Fully Convolution Networks for Semantic Segmentation）取得的图像分割效果明显超过以往传统的图像分割法。以此为标志，打开了深度学习在图像分割领域应用的新通道。全卷积网络 FCN 由多个卷积层和池化层组合而成，学习图像中每个像素的语义信息，最后生成像素级的类别概率预测图。

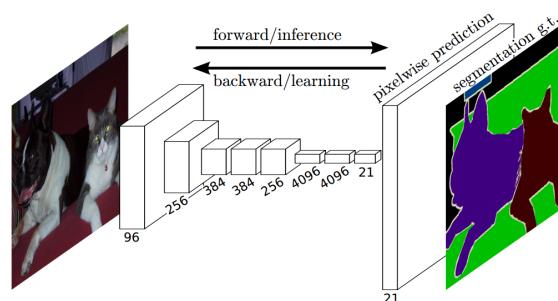


图 1–5 FCN 网络的结构<sup>[13]</sup>

Figure 1–5 The structure of Fully Convolution Networks

但由于下采样路径图像分辨率显著被降低，使得最后的预测结果比较粗糙。正如图1-5中分割结果中只能看到2个物体的轮廓，无法看出是一只猫或是一只狗。在此基础上，Hyeonwoo Noh等人<sup>[20]</sup>添加进去了上采样路径，亦即反卷积 Deconvolution，同步加入反池化 Unpooling，将图像分辨率逐步恢复到原来的分辨率。

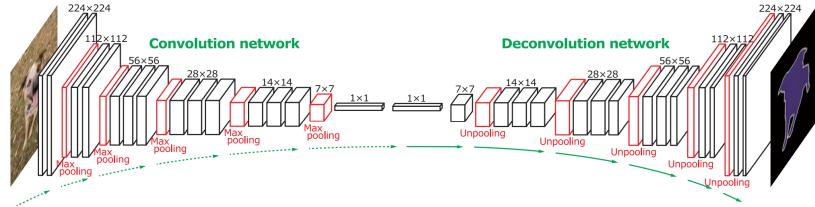


图 1-6 卷积下采样路径 vs. 反卷积上采样路径<sup>[20]</sup>

Figure 1-6 Convolutuion in down-sampling path vs. Deconvolution in up-sampling path

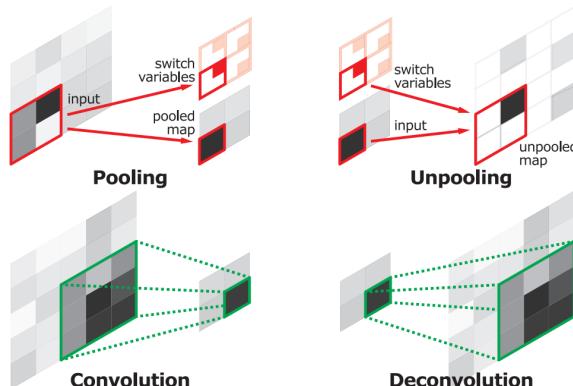


图 1-7 卷积与反卷积，池化与反池化的操作<sup>[20]</sup>

Figure 1-7 The operations of convolution/deconvolution, and pooling/unpooling

图1-6中的 Convolution network 与 Deconvolution network 分别组成下采样路径和上采样路径，这两条对称路径构成了图像语义分割网络架构。

后来，Olaf Ronneberger 等人<sup>[14]</sup>则进一步改进，保留下采样路径用于提取深度特征信息，而在上采样路径则加入了跳跃连接，将下采样获取的深度特征信息跳跃过来与上采样拼接起来，实现特征图融合。这就是经典的 U-Net 网络结构，如图1-8所示，U-Net 网络实现了更精细的图像语义分割结果。

U-Net 网络目前已成为医疗图像分割任务的基础网络，有很多研究人员在 U-Net 基础上改进，应用在医疗图像的分割。比如肝脏肿瘤、肺癌结节、乳腺肿块等不同的疾病。Li Xiaomeng 等人<sup>[21]</sup>提出一种基于 U-Net 混合式的稠密连接的新型网络结构

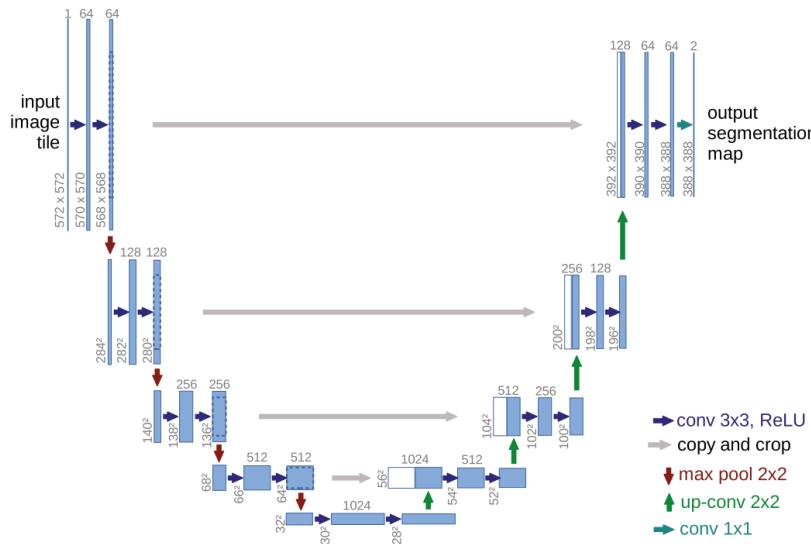
图 1-8 经典的图像分割 U-Net 网络结构<sup>[14]</sup>

Figure 1-8 The classic U-Net architecture for image segmentation

H-DenseUNet 用于肝癌的 CT 图像的分割。它保留了 UNet 低卷积层的细节特征，UNet 网络越来越深的层次增加了训练的长耗时，而 H-DenseUNet 又可以解决这些深层网络的长耗时训练困难问题。

医疗 CT 扫描图像基本上都是三维体数据，基于此又引申发展出 3D CNN，Jose Dolz 等人<sup>[22]</sup>就提出了 3D Fully convolutional networks，Özgün Çiçek 等人<sup>[17]</sup>则提出 3D-UNet 用于稠密的体数据分割。如此众多的分割网络，各个作者在 UNet 的基础上加上自己的创新，形成一个个独具特色的新型网络结构。那有没有一种广泛通用的网络来做图像分割呢？Mathias Perslev 等人<sup>[23]</sup>就提出了这样的一个想法和实施方案 One Network to segment them all 一种通用的轻量级的网络结构，如图 1-9 所示，来精确分割 3D 图像。它是在 2D UNet 网络之后再加上一个 Fusion model 实现的。

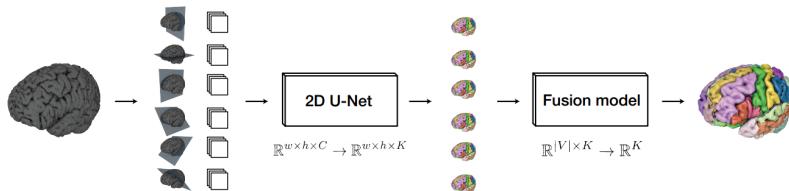
图 1-9 一种通用的轻量化的 UNet 网络<sup>[23]</sup>

Figure 1-9 A general, lightweight UNet to segment them all

最近三年，新冠肺炎疫情肆虐全球，为了抗击疫情，许多研究人员的兴趣都被 COVID-19 吸引过去，争相研究感染了新冠病毒的肺部 CT 影像，以期帮助医疗专家、临床医生了解被感染后的肺部症状，预测是否感染了新冠肺炎。Zahid Ullah 等人<sup>[24]</sup>提出一种稠密型注意力机制的网络来检测 COVID-19 的感染情况。其网络结构（图 1-10）在卷积层后插入通道注意力层和稠密区块层，使其聚焦于感兴趣的感染区域，从而高效地检测出感染情况。通过跟正常肺部影像和普通肺炎所造成阴影区域对比，比较明

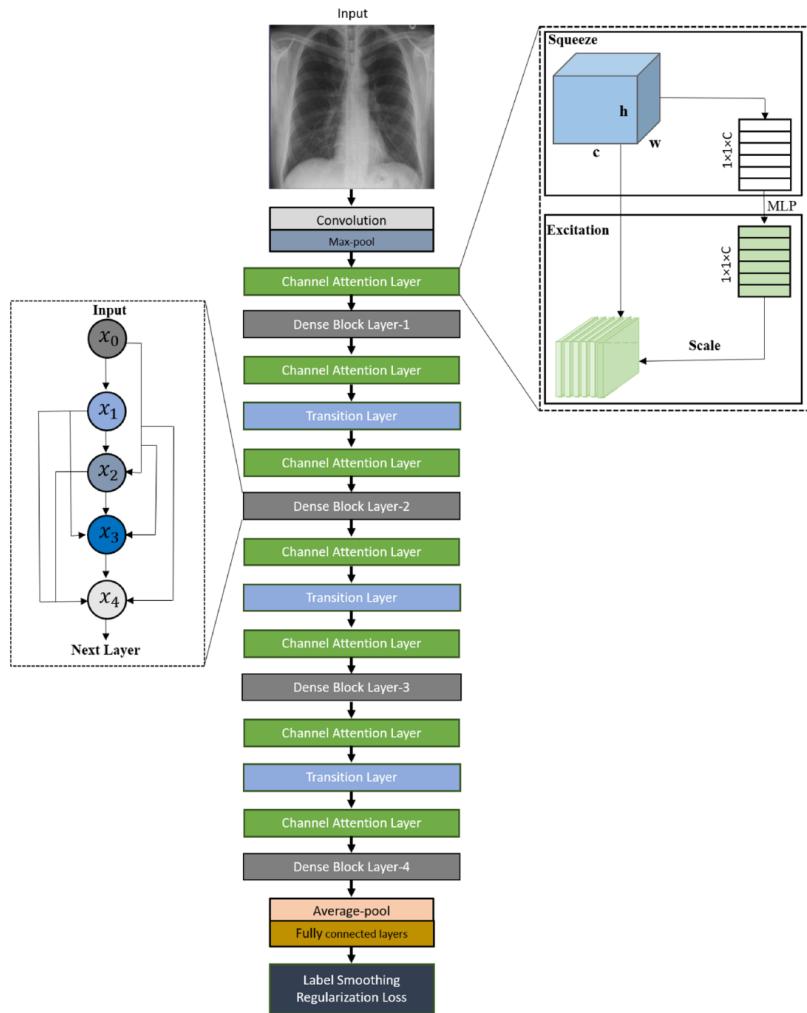


图 1-10 稠密型注意力机制网络用来检测 COVID-19 的感染情况<sup>[24]</sup>

Figure 1-10 Densely attention mechanism network to detect the COVID-19 infection

确地指出 COVID-19 感染区域（图 1-11 中 COVID-19 一栏所高亮显示的），展现给临床医生，辅助诊断是否感染新冠肺炎。

最新的进展是，随着 NLP 的 Transformer 模型<sup>[25-26]</sup>取得成功后，受此启发，NVIDIA

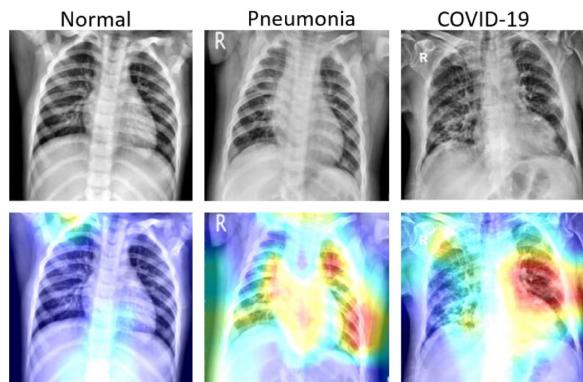


图 1-11 对比正常肺部、普通肺炎和 COVID-19 新冠肺炎的阴影，突出新冠肺炎感染区域<sup>[24]</sup>

Figure 1-11 Highlight the COVID-19 infection area by comparing with normal and pneumonia

的 Ali Hatamizadeh 联合 Vanderbilt University 的 Yucheng Tang 等人<sup>[27]</sup>率先将 Transformer 引入 3D medical image segmentation 中来, 提出了 UNETR 网络结构(见图1-12所示), 实现了 State-of-the-art (SOTA) 医疗图像分割性能新记录。

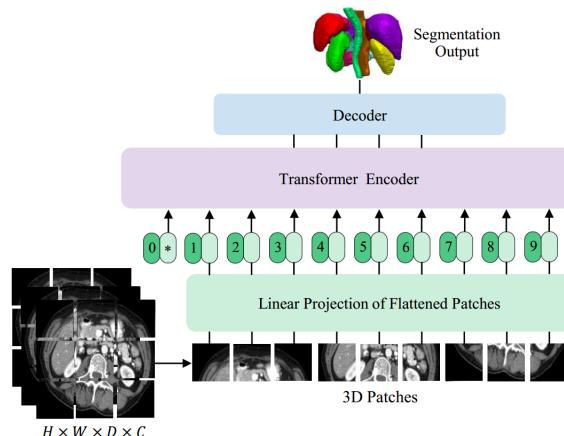


图 1-12 UNETR 网络基本结构<sup>[27]</sup>

Figure 1-12 The basic structure of UNETR network

综上所述, 基于深度学习的医疗图像分割技术发展非常活跃, 研究进展迅速, 进步巨大。

### 1.3.3 针对肺部支气管气道树的医疗图像分割方法

对于肺部 CT 图像的分割, 卷积神经网络 CNN 早就应用在上面了。最知名的 Lung Nodule Analysis 2016 大挑战竞赛要求参赛选手分割出 Lung CT 图像的肺结节,

判断肺结节是良性的还是恶性的，从而筛选出肺癌疑似患者，进行早期预防和针对性治疗。从 LUNA16 大挑战竞赛里诞生了两个具有重要影响力的公开数据集 LUNA 和 LIDC-IDRI<sup>[28]</sup>。类似于 LUNA 大挑战竞赛，针对肺部支气管气道树的分割也有一个比较著名的 EXtraction of Ariway from CT 2009 (EXACT'09) 竞赛项目。Pechin Lo 等人<sup>[29]</sup>在 2009 年发起了 EXACT'09 这个竞赛项目，其研究目标就是提供一个公开通用的数据集，鼓励参赛选手开发出创新性算法，从胸部 CT 扫描图像中提取出支气管气道树，评比参赛选手们提交的算法。

自 EXACT'09 大挑战竞赛的激励后涌现了大量的研究成果，Zhao Tianyi 等人<sup>[30]</sup>开发出来一个两阶段 2D + 3D 的神经网络和一个基于线性规划的气道分割跟踪算法。之后 2D CNN 和 2.5D CNN 网络相继被提出来，Yun Jihye 等人<sup>[31]</sup>提出的 2.5D 卷积网络进行逐个体素分割。在轴向 (Axial)、矢状 (Sagittal) 和冠状 (Coronal) 三个正交方向上取相邻的三个切片，送入 2 个卷积层的网络，然后将他们合并起来。这个方法在分割粗支气管时效果还不错，减少了假阳性，检测到的气道树的长度也增长了不少。Qier Meng 等人<sup>[32]</sup>直接采用了 3D FCN 网络，沿着气道的中心线跟踪气道，并根据每个气道的直径和运行方向设置 VOI (感兴趣的立方体体积)，3D FCN 提取 VOI 内的气道树，最后将所有的 VOI 区域内的气道合并为一个完整的气道树。浙江大学医学院第二附属医院的 Wu Xiaoming 等人<sup>[33]</sup>在 UNet 网络结构的基础上设计了一个 Tiny Atrous 卷积网络，称为 TACNet<sup>[33]</sup>结构（见图 1-13），并在 EXACT'09 数据集上对平均 Tree Length Detected 取得了 SOTA 性能。

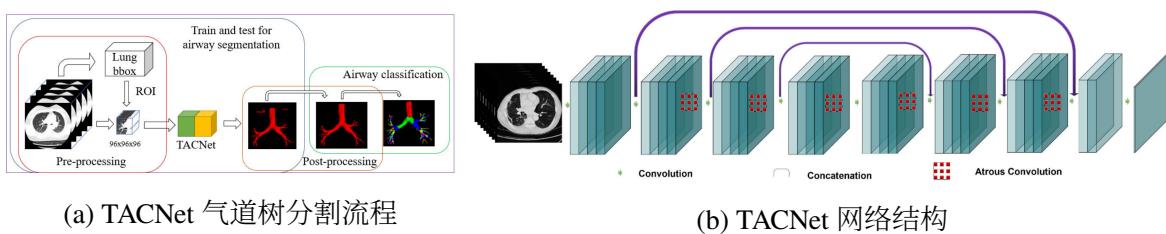


图 1-13 TACNet 气道树分割网络<sup>[33]</sup>

Figure 1-13 Tiny Atrous Convolution network to segment the airway tree

以上一些研究成果是基于基础的 CNN 卷积网络实现了对支气管气道树的分割，在当时提出来时都具有一些创新性，但对分割的精细程度显得不足。他们一般能分割到叶支气管 (Lobar bronchus)，但对于更纤细的段支气管 (Segmental bronchi)、末端小叶支气管 (Lobular bronchi) 和延伸到肺泡的细支气管 (Bronchiole)，他们已经显得能力不够，难以分割更细小的支气管了。

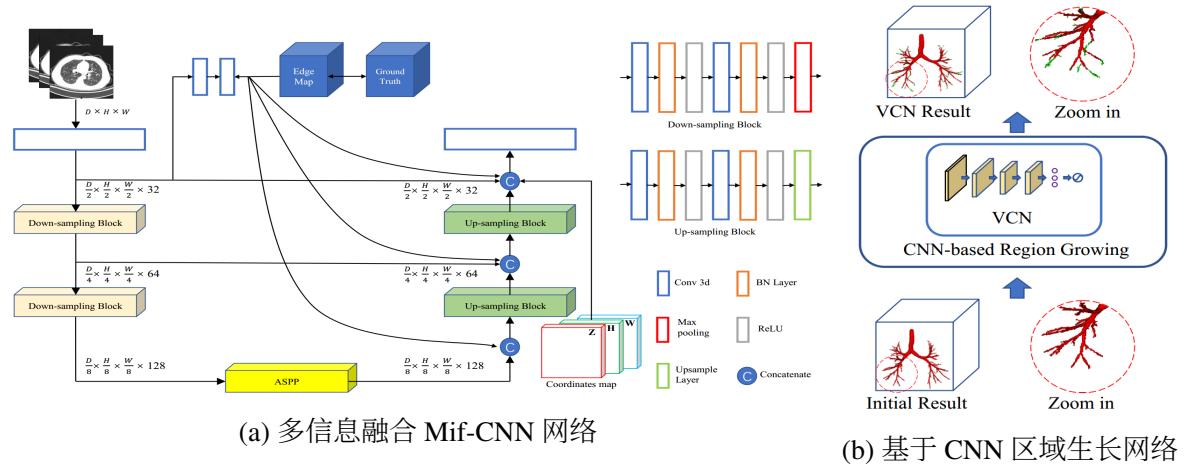


图 1-14 多信息融合 Mif-CNN 网络和基于 CNN 区域生长网络的组合<sup>[34]</sup>

Figure 1-14 Corse-to-fine airway segmentation, Mif-CNN and CNN-based region growing combination

随着深度学习图像分割技术的发展，更新的技术、更强大的网络结构被开发出来。Guo Jinquan 等人<sup>[34]</sup>利用多信息融合网络（图1-14a所示）和基于 CNN 的区域生长网络（图1-14b）相互结合的方式，由粗到细迭代式逐渐分割出更细小的支气管气道树。

肺部支气管气道是一棵树的形状，从 CT 图像中提取支气管树可视为支气管镜手术导航的细粒度图。上海交通大学医疗机器人研究院的 Yu Weihao 等人<sup>[35]</sup>将图神经网络的概念迁移过来，创新性地提出了一种基于图神经网络的结构，并命名为树网络（Tree Neural Network, TNN<sup>[35]</sup>）。这种树网络结构巧妙地将支气管分支看成是映射到图空间的节点，提取气道树视为构建节点之间连接的边，也就是支气管分支连接的超边（Hyperedges），见图1-15所示。

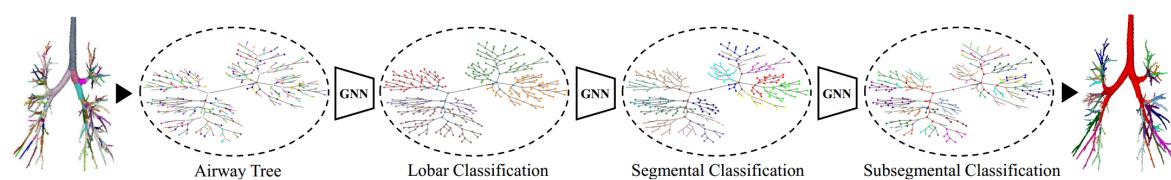


图 1-15 创新性的树网络结构，用 GNN 来提取支气管气道树<sup>[35]</sup>

Figure 1-15 The novel Tree Neural Network, extract the airway tree by virtue of GNN

这种创新性的树网络结构带来了很大的性能提升，尤其对于分割叶支气管、段支气管和小叶支气管的分割分别能达到 93.6% 和 82.0%<sup>[35]</sup>的精确度。

## 1.4 挑战和困难

尽管现有的一些研究取得很优秀的成果，肺部支气管气道树的分割性能不断提升，但仍有非常大的挑战需要克服。主要表现在如下方面：

- **类别严重不平衡**

像肺部支气管、血管、动静脉等管状物体，在CT扫描的切片图像中，前景像素（指支气管管腔、管壁的像素）与背景像素（指除了支气管管腔管壁的像素之外的像素）个数对比差别巨大，这就造成严重的类别不平衡。而绝大部分卷积网络都严重依赖标注数据，标注数据缺乏，高质量的标注数据更是稀少，这将导致分割效果大打折扣。肺部树状的支气管、肺动脉和肺静脉都是非常纤薄的，在肺部都是发散地分布着。这完全不同于肝脏、肾脏和胰腺这些块状器官<sup>①</sup>，标注这些块状器官的前景像素(Pixel)或体素(Voxel)<sup>②</sup>与背景像素或体素相差不甚巨大，但对于肺部支气管，标注的像素或体素个数与背景像素或体素个数相比则差距悬殊。在单张切片  $H \times W = 512 \times 512$  这样的高扫描分辨率下，叶支气管 Lobar bronchus、段支气管 Segmental bronchi 分别可能只有 4 个，3 个到 2 个像素，而小叶支气管 Lobular bronchi 则只有 1 个像素甚至不到 1 个像素。这种毛细的管状物不仅加剧了精细标注的难度，也导致标注的前景像素与背景像素对比，强烈不平衡。

若卷积网络学习一张  $H \times W = 512 \times 512$  像素的灰度图像，而标记数据只有两、三个像素。即使将卷积核尺寸(Kernel size)调整到最小，移动步长(Stride)调整到最小，最后得到的分割结果其假阳性率(False Positive Rate)也是非常高，精度也是非常差。尽管有加权交叉熵损失 Weighted cross-entropy loss<sup>[36]</sup>函数和数据增强 Data Augmentation 技术，仍难以保证有效地学习到图像的特征信息。

- **全局尺度和局部尺度的上下文信息难以同时兼顾**

肺部支气管是在三维空间上分布的，其众多的分支分散在肺部各处。我们的分割当然不仅要学习到气道(Trachea)、左右主支气管(Left/Right main bronchus)、上/中/下肺叶支气管(Superior/Middle/Inferior Lobar bronchus)的特征信息，还要学习从上/中/下肺叶支气管扩展出来的段支气管(Segmental bronchi)，从段支气管扩展出来的小叶支气管(Lobular bronchi)，甚至连接到肺泡的细支气管

① 块状器官指的是肝脏、肾脏和胰腺在CT图像上看起来是一个大的区块，非常易于辨认，也容易做区块边缘分割。

② 注：像素 Pixel 指在平面图像中一个高×宽为( $H \times W = 1 \times 1$ )的平面网格，体素 Voxel 指的是在 CT 这样的三维图像上一个深×高×宽为( $D \times H \times W = 1 \times 1 \times 1$ )的立方体格，其中  $D = 1$  指的是一个轴向切片的厚度。

(Bronchiole) 的特征信息。细小的支气管这种局部尺度的上下文其实不需要多大的感受野，但对于粗大一些的气道、左右主支气管，我们又需要更大一些的感受野来推测整体的气道树形状，也即全局尺度的上下文。面对这种矛盾，应该使用多少个池化层？如果简单地堆积更多的池化层，那么增加的参数可能会因为训练数据不足而导致过拟合。如果为了避免这种参数爆炸而牺牲卷积网络的宽度（即特征通道的数量）来换取深度（卷积层的层数），那么模型的学习和拟合能力可能不够<sup>[37]</sup>。所以，全局尺度的上下文信息与局部尺度的上下文信息是否能够兼顾到呢？这是另一个困难与挑战之处。

## 1.5 本文的研究内容和创新点

本文针对肺部 CT 扫描三维图像，利用基于深度学习的三维卷积网络 3D-UNet 模型<sup>[14]</sup>来分割提取人体肺部支气管树状的气道结构。我们使用来自 MICCAI2022 的 Airway Tree Modeling(ATM) Challenge 2022(以下简写为 ATM22<sup>[38-41]</sup>) 的数据集，先将 NIfTI 格式的 CT 扫描三维图像裁切成长方体，只保留肺部区域。我们以 3D-UNet 作为基准模型，进行基准实验，观察 3D-UNet 基准模型对支气管气道树的分割效果。研究 3D-UNet 分割模型的不足之处，我们提出了改进方案。本文的创新之处如下：

1. 引入损失函数 Focal Loss，重塑常用的交叉熵损失来解决类别严重不平衡问题

由于肺部支气管标记数据的稀疏性，造成前景标记像素与背景像素对比严重的类别不平衡。常规的交叉熵 Cross-Entropy Loss 损失函数在训练过程中遇到极端的前景和背景类别不平衡时，分类器对于大量的无分类数据，其权重分配会越来越向这种无分类数据倾斜，而真正拥有分类的数据其获得的权重分配几乎被淹没了，所以导致分类精度下降严重。Focal Loss 损失函数则是重塑交叉熵损失函数，使权重分配向有分类的数据倾斜，从而提高分割的精度。

2. 引入注意力蒸馏方法，增加额外的梯度，聚焦于辨识更细小纤薄的支气管

自 NLP 研究中提出注意力机制<sup>[26]</sup>后，我们认识到注意力图是在指导网络聚焦看图像的一些重要的地方，这些重要的地方是区分或辨识物体的显著特征。这种有价值的注意力图可以从教师网络逐层传递到学生网络，从知识蒸馏的角度说，就是知识是可传递的。教师网络学习到的知识可以传递到学生网络。我们就是在注意力机制的启发下，将注意力蒸馏机制方法引入到我们的 3D-UNet 网络，在上采样路径给相邻的两个反卷积层增加一个额外的梯度，使网络聚焦于辨识更细小纤薄的支气管，如段支气管和小叶支气管。

3. 为改善池化层对支气管空间位置信息的捕捉, 设计一个通道级特征再学习模块, 用于强化与任务相关的特征, 即关键的空间位置信息

图像分辨率在下采样路径池化层收缩, 在上采样路径池化层膨胀, 支气管空间位置信息在不同的分辨率下没有被区别对待, 池化层没有对支气管空间位置信息有效地捕捉。而这些关键的空间位置信息对于支气管气道树的精细分割至关重要, 因此为了强化与任务相关的特征, 为了很好地捕捉支气管的空间位置信息, 特地设计一个通道级特征再学习模块。

## 1.6 论文组织结构

论文的各个章节安排如下:

### 第一章：绪论

介绍本论文的研究背景和意义, 追踪关于肺部支气管气道树分割这个课题的研究现状和发展趋势。提出本论文的研究内容和有哪些创新点。

### 第二章：基准分割模型 3D-UNet 和分割效果分析

阐述 3D-UNet 基准模型的原理, 设计实验对支气管气道树进行分割, 研究与展示实验结果, 分析不足之处。

### 第三章：注意力蒸馏方法对基准模型的改进

阐述注意力蒸馏的原理, 取上采样路径四个卷积层的输出结果, 进行注意力蒸馏操作, 然后对比研究其对分割效果的改进作用。

### 第四章：通道级特征再学习方法提取精细支气管特征

阐述通道级特征再学习的基本原理和计算方法, 实现特征再学习模块。将特征再学习模块加到 3D-UNet + AD 网络进行训练, 研究训练的结果, 分析评估其对支气管气道树分割的改进效果。通过一个综合实验展示综合分割效果和指标参数, 在验证集和测试集上分割出精确完整的三维气道树结构。

### 第五章：总结与展望

对本文所做的工作进行总结, 并对肺部支气管气道树分割这个课题的未来研究方向进行展望。

需要指出的是, 本文会穿插着进行对比实验, 没有专门的章节来讲解消融研究 Ablation Study, 消融研究将体现在对比实验中。



## 第二章 基准分割模型 3D-UNet 和分割效果分析

### 2.1 医疗图像语义分割网络层次

图像的语义分割是一种像素级的分类技术，输入一张（RGB 彩色或灰度）图片，分类器输出图片上每一个像素所属的类别标签。基于深度学习的图像语义分割方法，其主要算法是卷积神经网络 (Convolutional Neural Network, CNN)。卷积神经网络通常包含以下四层：

- 卷积层 Convolution layer
- 归一化层 Normalization layer
- 激活函数层 Activation function
- 池化层 Pooling layer

CNN 的基本网络结构如图2-1所示：

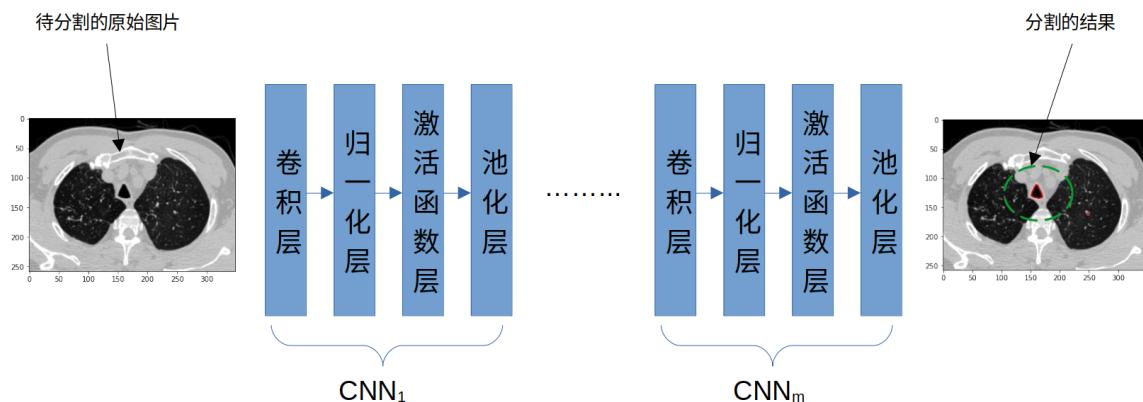


图 2-1 卷积神经网络基本构成  
Figure 2-1 The basic structure of CNN

待分割的原始图片经过一层或连续多层的卷积网络后，图片的低层位置信息与高层语义信息等被提取出来，最后得到分割的结果。每次迭代将预测的结果与真实的分割结果进行对比，选取一个合适的损失函数，计算损失值。然后通过梯度下降的反向传播，对卷积网络的参数—也就是网络中每个神经元的权重 Weight 与偏置 Bias—进行更新。循环进入下一次迭代，直到损失值不再降低趋于平缓，网络收敛，整个训练过程完成。

### 2.1.1 卷积层

卷积层的主要作用是提取特征，获得图像的局部信息。其工作原理是：在输入图像上滑动  $H \times W = 3 \times 3$  的窗口，每滑动 1 个单位长度（此处滑动的长度称为步长 Stride）就提取这个  $3 \times 3$  窗口内的像素信息，与同样尺寸的一个权重矩阵（亦即卷积核尺寸 convolution kernel size）做点积相乘并求和，得到当前位置的特征。

令  $H3 \times W3$  卷积核权重矩阵  $W_{conv.kernel}$

$$W_{conv.kernel} = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \quad (2-1)$$

CT 扫描图像在  $H3 \times W3$  窗口内的 Hounsfield unit<sup>[42]</sup> 灰度<sup>①</sup> 像素矩阵  $X_{hu.gray}$

$$X_{hu.gray} = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \quad (2-2)$$

则当前位置的特征  $feat(X)$  为

$$\begin{aligned} feat(X) &= \sum (W \cdot X) + b \\ &= \sum \left( \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \right) + b \end{aligned} \quad (2-3)$$

其中  $b$  为偏置参数 Bias

式2-3是在二维图像平面上的卷积操作，若在 CT 三维体数据上进行卷积运算，则卷积核尺寸扩展为  $D3 \times H3 \times W3$ ，相应地 Hounsfield unit 灰度像素矩阵也同步扩展到三维。三维卷积的操作如图2-2所阐述。

### 2.1.2 归一化层

训练深度神经网络的计算强度非常大，训练时间漫长。减少训练时间的一种方法就是使网络中的神经元的活动归一化<sup>[44]</sup>。归一化是沿着某个指定的维度计算特征的均值与方差，对于推动深度网络收敛有着非常重要的作用。我们常用的归一化有：

<sup>①</sup> Hounsfield unit 亨氏单位是放射科医生在解释 CT 图像时使用的相对定量的无线电密度测量单位。CT 重建时使用身体组织对辐射的吸收/衰减系数来生成灰度图像。身体组织的物理密度与 X 射线束的吸收/衰减成正比。Hounsfield 单位，也称为 CT 单位，是根据 X 射线束的线性衰减系数进行线性变换计算而来的。

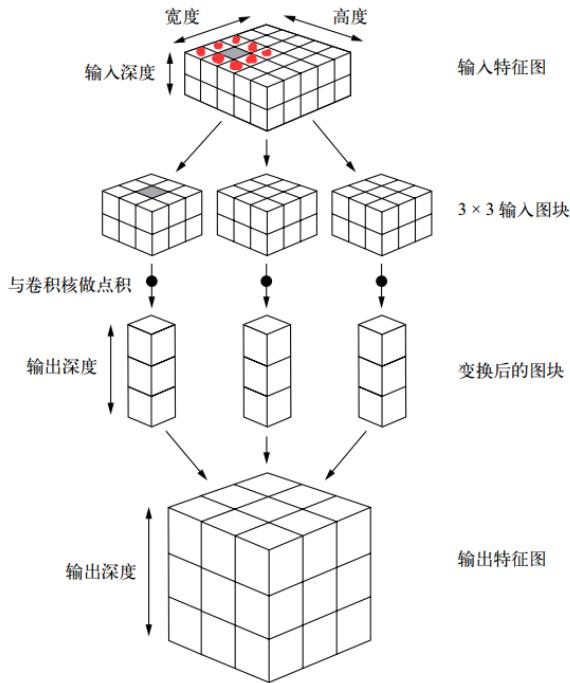
图 2-2 三维卷积操作的原理<sup>[43]</sup>

Figure 2-2 The principle of 3D convolution

### 1. 批次归一化 Batch Normalization<sup>[45]</sup>

批次归一化是沿着 Batch 维度计算特征的均值和方差来进行归一化，批次的大小 Batch size 决定着预测的误差。通常取更大一些的 Batch size 有利于降低误差。过小的 Batch size 会导致性能下降。

### 2. 层次归一化 Layer Normalization<sup>[44]</sup>

层次归一化是沿着 Channel 维度计算特征的均值和方差来进行归一化。因为批次归一化依赖于 Batch size，尤其是不能低于 mini batch size，而层次归一化是为了克服此问题的。

### 3. 实例归一化 Instance Normalization<sup>[46]</sup>

实例归一化跟批次归一化执行相同的计算，但却是对单个样本而言。实例归一化可用于防止特定实例的均值和协方差偏移，简化学习过程。

### 4. 群组归一化 Group Normalization<sup>[47]</sup>

群组归一化是把 Channel 切分为组，在每个组内计算特征的均值和方差。群组归一化被 Wu Yuxin 等人<sup>[47]</sup>提出来用于替换批次归一化，不同于层次归一化与实例归一化的方式，解决批次归一化对 Batch size 依赖的问题。

我们可以用图2–3来讲解这4种归一化方式的差别。深度网络的数据维度一般是

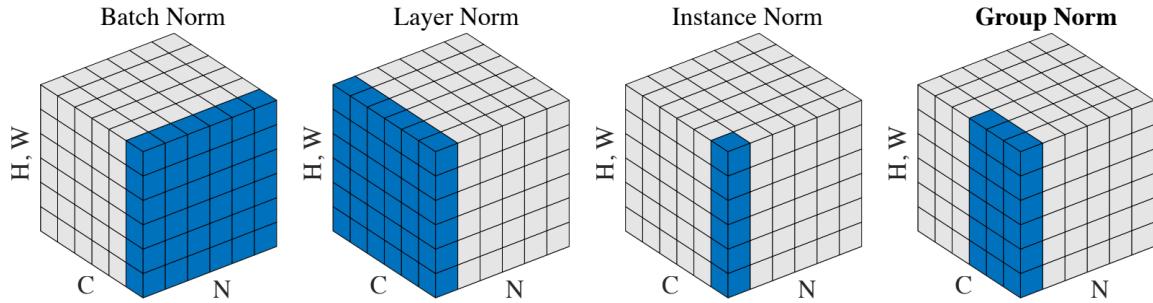


图 2–3 四种归一化方法<sup>[47]</sup>

Figure 2–3 Four Normalization Methods

$[Batch, Channel, Height, Width]$ , 简写为  $[N, C, H, W]$  ( $N$  即为  $Batch$ )。因为这是四个维度, 我们无法画出一个四维空间的图形, 所以压缩特征的  $H, W$  至一个维度。

批次归一化 Batch Normalization 沿  $Batch$  维度计算特征的均值和方差, 归一化为  $[N, H, W]$  的维度, 如图2–3 的 Batch Norm 子图的蓝色方格所示。

层次归一化 Layer Normalization 避开  $Batch$  维度, 而是沿  $Channel$  维度计算特征的均值和方差, 归一化为  $[C, H, W]$  的维度, 如图2–3的 Layer Norm 子图的蓝色方格所示。

实例归一化 Instance Normalization 选择单个样本计算特征的均值和方差, 单个样本 (也即一张图片) 的维度只有  $Height$  与  $Weight$ , 所以归一化  $[H, W]$  的维度, 如图2–3的 Instance Norm 子图的蓝色方格所示。

群组归一化 Group Normalization 则是介于层次归一化和实例归一化之间, 其将  $Channel$  维度切分为很多组  $Group$ , 在每一个组内计算特征的均值和方差, 归一化为  $[C//G, H, W]$  的维度, 如图2–3的 Group Norm 子图的蓝色方格所示。

此外还有权重标准化 (Weight Standardization<sup>[48]</sup>)、批次-通道归一化 (Batch-Channel Normalization<sup>[49]</sup>)、深度归一化 (DeepNorm<sup>[50]</sup>) 三种归一化方法, 在此不做深入介绍。

### 2.1.3 池化层

池化的作用是缩小图像在高度和宽度方向上的空间运算, 是一种降维的操作, 可以用来改变当前层的输出维度, 增大感受野。池化有两种方式: 最大化池化 Max Pooling 和平均池化 Average Pooling。最大化池化是从目标区域 (图2–4的红色区域) 中取出最大值, 平均池化则是计算目标区域 (图2–4的深蓝色区域) 的平均值。

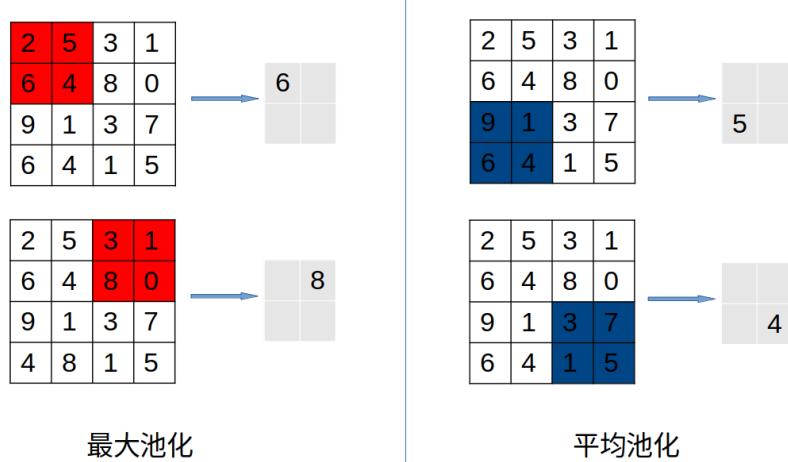


图 2-4 池化的基本操作

Figure 2-4 The basic operations of pooling

池化层具有如下 2 个特征：

### 1. 没有要学习的参数

池化层与卷积层不一样，它没有要学习的参数。池化只是计算目标区域内的最大值或者平均值，所以不存在参数需要学习而改变。

### 2. 卷积的通道数保持不变

经过池化层，输入数据跟输出数据的通道数保持不变，是按照通道独立计算的，如图2-5所示。

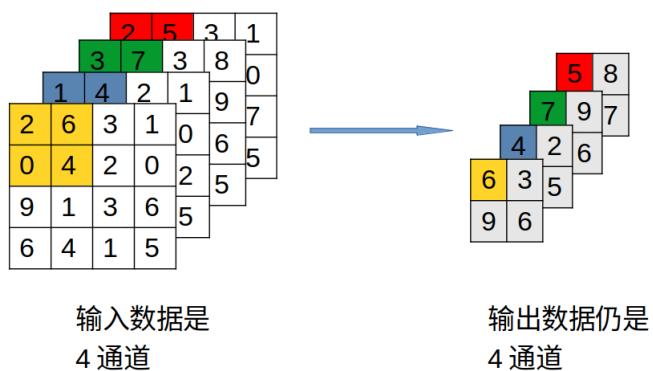


图 2-5 池化层按通道独立计算，通道数保持不变

Figure 2-5 Pooling layer will keep the same number of channels

## 2.2 3D-UNet 基准模型

本文我们采用 3D-UNet<sup>[17]</sup>网络结构模型来分割支气管气道树，3D-UNet 网络是将经典的 UNet<sup>[14]</sup> 网络从平面图像分割扩展到三维体数据的。跟 UNet 网络的 U 型结构相似，我们将卷积层换成 3D 卷积，归一化层换成 3D 归一化，相应地池化层也更换为 3D 池化。CT 图像是一种体数据形式，它由一层一层的切片堆叠而成，每一层切片是一张二维的灰度图像。相比于平面图像中的像素，在 CT 图像中我们定义体素 Voxel 为  $D \times H \times W = 1 \times 1 \times 1$  的立方体所含的像素， $D = 1$  表明是一层切片。3D-UNet 网络的输入是  $D \times H \times W$  的体数据，我们称之为 Cuboid，输出是每个体素的二分类分割概率图。我们在下采样路径设计 4 个卷积层块，每个卷积层块包含 2 层卷积。而在上采样路径同样设计 4 个反卷积层块，每个反卷积层块包含 2 层反卷积。

### 2.2.1 网络结构设计

沿着下采样路径，每个卷积层块的结构如图2-6所示，将输入的体数据图像的通道数扩增一倍，从输入的  $n$  个特征通道扩增到  $2n$  个特征通道。经过 3D 最大池化后，分辨率降低一半。

沿着上采样路径，每个反卷积层块的结构如图2-7所示，将输入的体数据图像的通道数缩减一半。从输入的  $n$  个特征通道缩减到  $\frac{n}{2}$  个特征通道。经过上采样池化后，分辨率升高一倍。将下采样路径降低的分辨率恢复起来，这样就实现端到端，体素到体素的分割。

我们的 3D-UNet 主干网络包括编码器 Encoder 和解码器 Decoder 两大部分，编码器主要用来实现图像的特征提取，扩大感受野，输出具有类别信息的高层语义特征。解码器则逐步恢复分辨率，从而提取到低层的位置信息。更重要的是，在编码器和解码器对应的层次之间，引入跳跃连接，将包含丰富细节信息的编码器特征拼接到解码器的高级分类特征信息中来，这样两种特征信息互相补充，最后输出每个像素的分类概率图。我们的 3D-UNet 网络结构详情如图2-8所示。由于图形横向宽度较大，所以将其横向放置。将 CT 体数据 (网络左侧) 送入网络之前，我们将其裁切为一个一个  $D80 \times H192 \times W304$  规格的长方体<sup>①</sup>，网络学习长方体的图像特征，分割长方体内的支气管分支。训练完成后我们将属于同一个原始 CT 图像的所有长方体拼接起来，最终分割出来完整的支气管气道树 (网络右侧)。在长方体上面标出来的数字表示经过卷积操作后输出的特征通道数，而在长方体侧面标注的诸如  $D20 \times H48 \times W76$  表示

<sup>①</sup> 我们将在后文讲述为什么裁切成  $D80 \times H192 \times W304$  规格的。

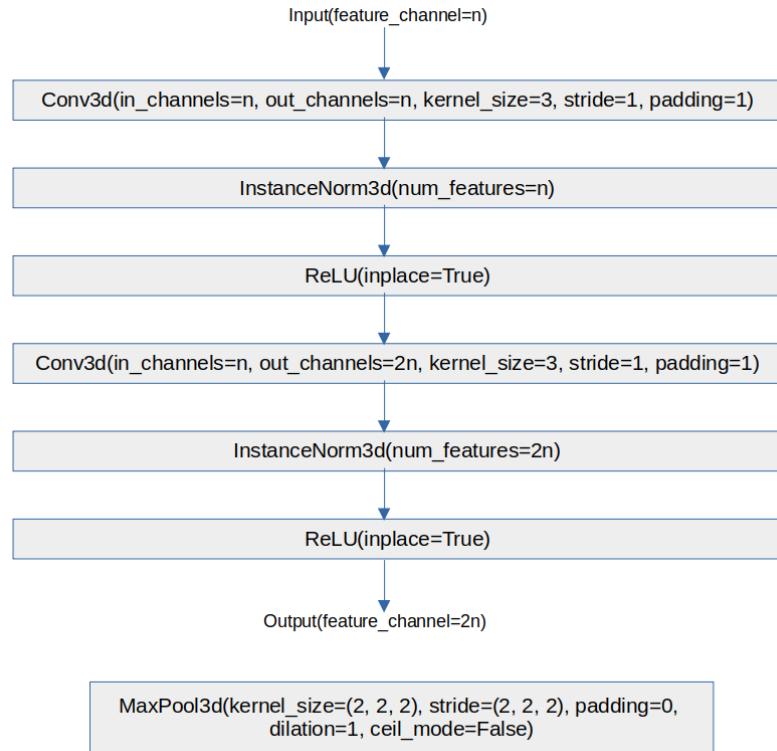


图 2–6 3D-UNet 下采样路径上卷积层块和池化层的结构

Figure 2–6 The convolution block and pooling layer in the down-sampling path of 3D-UNet

经过池化后，长方体的体积扩大或缩小，也就是长方体内体素的分辨率增大或减小。需要指出的是，在 U 型结构的右侧，每一个层靠近跳跃连接的箭头处都有一个并排的长方体柱，那表示跳跃连接将下采样路径上每一个卷积层块的特征输出拼接到上采样路径上每一个对应的反卷积层块的输入特征来。

上述的 3D-UNet 网络结构是本文所设计与使用，并编程实现的 CT 图像支气管气道树分割网络。我们将以此 3D-UNet 网络作为基准<sup>①</sup>，通过实验分析 3D-UNet 的分割结果和性能指标（包括但不限于假阳性率 False Positive Rate、真阳性率 True Positive Rate、骰子相似度系数 Dice Similarity Coefficient、分支检出率 Branch Detected、检测到的树长 Tree Length Detected 和精度等指标），提出我们的改进方法。我们将在后面的章节展开讲述我们的改进方法，并通过实验来对比验证我们的改进方法是否有效，是否提高了分割效果和性能。

<sup>①</sup> 需要指出一点，本文后面所提到的 3D-UNet 网络基准模型，3D-UNet 基准网络或基准模型都指的是该 3D-UNet 网络，它们是在不同语境下的叫法。

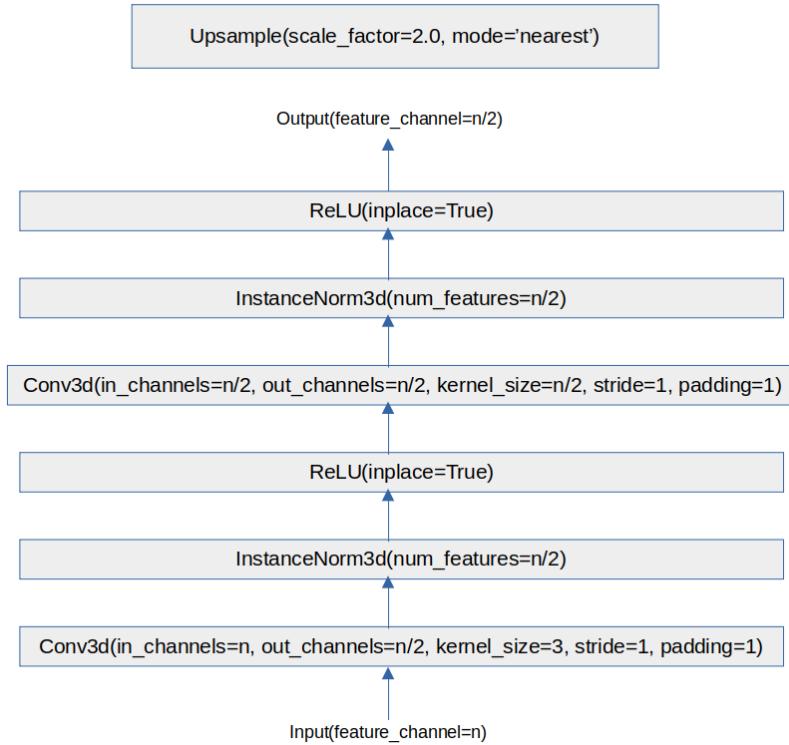


图 2-7 3D-UNet 上采样路径上翻卷积层块和池化层的结构

Figure 2-7 The deconvolution block and pooling layer in the up-sampling path of 3D-UNet

## 2.2.2 损失函数、优化器与学习率调整

### 2.2.2.1 Dice Loss 损失函数

我们为 3D-UNet 网络基准模型采用普遍的骰子损失函数 Dice Loss, 其计算公式2-4

$$L_{dice}(Cuboid_{pred}, Cuboid_{gt}) = 1 - \frac{2 \sum (Cuboid_{pred} \cdot Cuboid_{gt}) + \epsilon}{\sum Cuboid_{pred} + \sum Cuboid_{gt} + \epsilon} \quad (2-4)$$

其中  $Cuboid_{pred}$  表示预测值三维矩阵,  $Cuboid_{gt}$  表示真实值三维矩阵, 此三维矩阵的大小为2.3 节中切割出来的长方体子块的大小  $z \times y \times x = 80 \times 192 \times 304$ . 而  $\epsilon$  是为防止除零而加入的平滑常数,  $Cuboid_{pred} \cdot Cuboid_{gt}$  表示这 2 个三维矩阵的点积。

### 2.2.2.2 Focal Loss 损失函数

一个肺部 CT 扫描图像中, 支气管体素占全部体素的比例是很低的, 从表2-1就可以看出, 前景体素(即支气管体素)与背景体素(即除支气管体素之外的体素)的比例分别达到了 5 : 1000、3 : 10000、9 : 10000、7 : 10000、6 : 10000, 这是一种严重

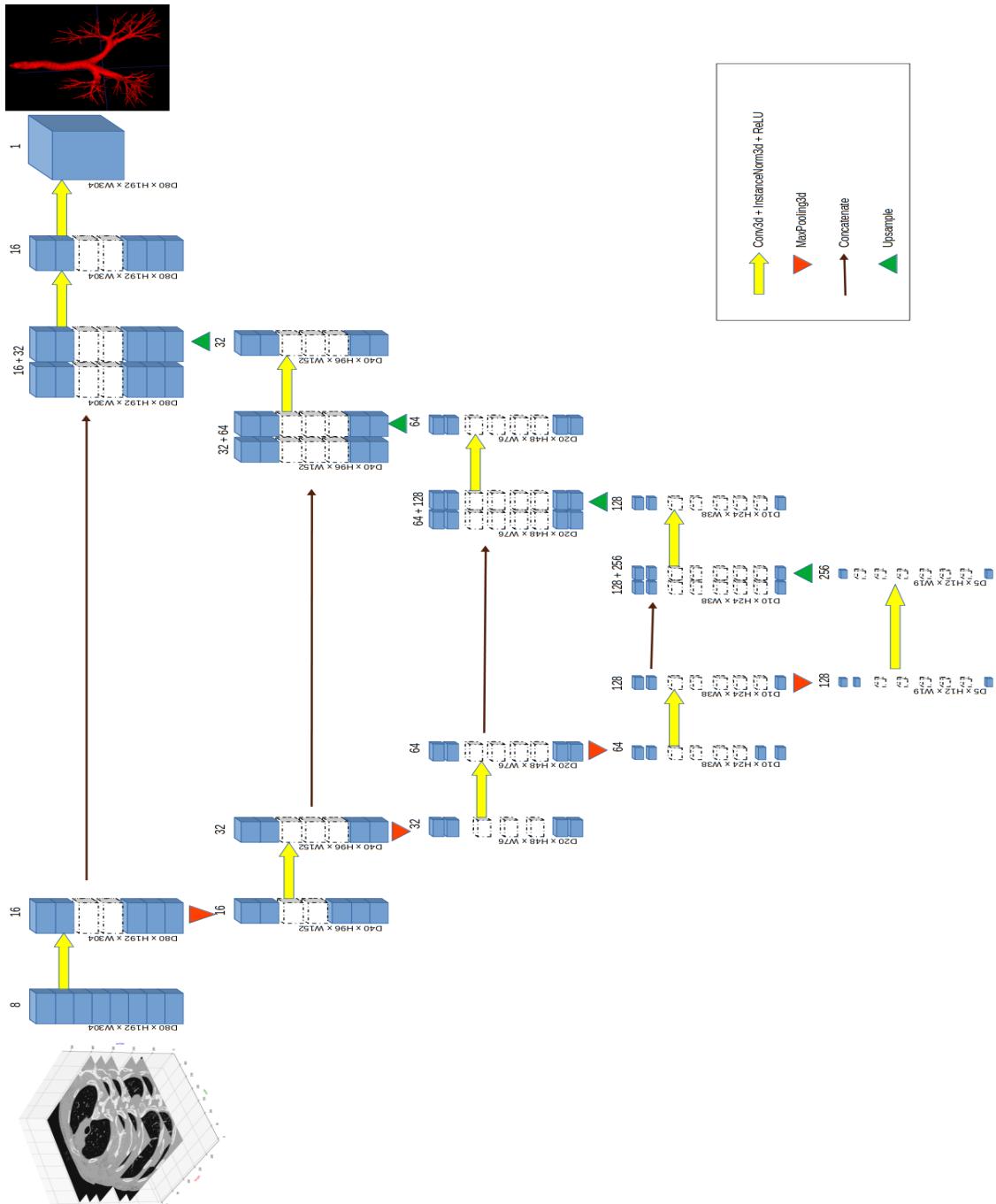


图 2-8 本文所设计与使用的 3D-UNet 网络结构

Figure 2-8 The 3D-UNet structure we designed in this thesis

的类别不平衡。正是这种严重类别不平衡，就会使得训练过程中损失会逐渐偏向背景体素，直至完全主导了损失，完全压过了前景体素对损失的贡献。类别不平衡会带来

如下问题：

1) 训练效率低下。因为绝大部分是无用的背景体素，没有贡献有用的学习信息。这也是为什么我们要对原始的 CT 图像进行裁切（见2.3小节），裁切掉大部分的黑色背景，只保留肺的最小包围框作为有效的区域。即使裁切掉了大部分背景区域，但前景体素占比仍然极端稀疏。

2) 这些无用的背景体素主导了损失，压倒了训练，导致模型退化。

为了对抗这种严重的类别不平衡，我们引入了 Focal Loss 损失函数<sup>[51]</sup>。

支气管气道树分割任务是属于二分类问题，即对肺部 CT 扫描三维体数据的每一个体素来说，要么属于支气管，要么属于背景。传统的二分类图像分割都采用交叉熵损失 Cross Entropy Loss 作为损失函数。令  $p$  为预测某一体素属于支气管体素的概率值， $gt$  为该体素的真实值，则交叉熵损失  $CELoss(p, gt)$  为：

$$CELoss(p, gt) = \begin{cases} -\log(p) & \text{if } gt = 1; \\ -\log(1-p) & \text{if } gt = 0. \end{cases} \quad (2-5)$$

概率  $p \in [0, 1]$ 。为表示方便起见，令

$$p_t = \begin{cases} p & \text{if } gt = 1; \\ 1-p & \text{if } gt = 0. \end{cases} \quad (2-6)$$

则交叉熵损失  $CELoss(p, gt)$  可简写为：

$$CELoss(p, gt) = CELoss(p_t) = -\log(p_t) \quad (2-7)$$

在交叉熵损失函数中，为了对抗这种类别不平衡问题，我们通常会引入一个权重因子  $\alpha_t$  将交叉熵损失函数改进为加权交叉熵损失函数 Weighted Cross Entropy Loss，也有论文称其为  $\alpha$  平衡交叉熵损失函数。

$$WCELoss(p_t) = -\alpha_t \log(p_t) \quad (2-8)$$

尽管加权交叉熵损失函数平衡了一些前景体素与背景体素不平衡的类别比例问题，但还不够，它仍然难以区分易于分割的（粗大）支气管体素和难以分割的（细微）支气管体素。何为易于分割的支气管体素，什么是难以分割的支气管体素？以图 2-9 中的枝干气管和末梢支气管为例来讲，同一个 ATM\_060\_0000 病例，在第 392 层切片处截取到的是粗大的气管，而在第 119 层切片处截取的却是细微的末梢支气管。很显

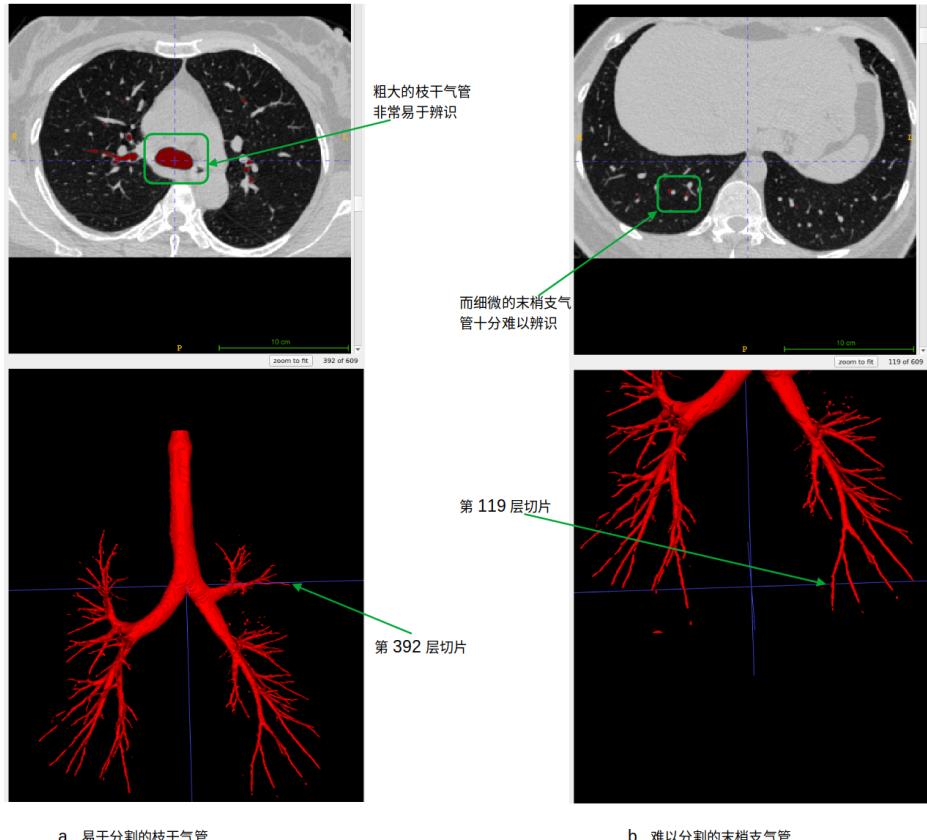


图 2-9 易于分割的枝干气管和难以分割的末梢支气管

Figure 2-9 Trunk trachea is easy to segment, but distall bronchus are quite hard to segment.

然，分割粗大的枝干气管体素比细微的末梢支气管体素要容易得多。因此在训练中我们需要做出一些调整，降低易于分割的体素的权重，提升难以分割的体素的权重，为了分割出更多的支气管体素，做出这种倾斜是十分必要的。基于这种考虑，我们对加权交叉熵损失函数引入一个调节因素  $(1 - p_t)^\gamma$ ，改变加权交叉熵损失为

$$WCELoss(p_t) \rightarrow FLoss(p_t) = -\alpha_t \cdot (1 - p_t)^\gamma \cdot \log(p_t) \quad (2-9)$$

这种损失函数称为 Focal Loss 损失函数。其中两个超参数  $\gamma$  称为聚焦因子， $\alpha_t$  称为平衡因子。

Focal Loss 损失函数最初是由 Facebook AI Research 的 Tsung-Yi Lin、Kaiming He 等人<sup>[51]</sup>提出来用于稠密的物体检测的新型损失函数，我们将其引入到支气管气道树分割任务这种存在严重类别不平衡的场景中来。其目的是为了对抗前景体素与背景体素的类别严重不平衡的问题，也为了在训练过程中调整（倾斜）权重给难以分割的

体素，使模型聚焦于有用的前景信息和难以分割的体素，从而帮助提升分割模型的精度和真阳性率（也称为灵敏度）等指标。

Focal Loss 损失函数的  $\alpha$ 、 $\gamma$  两个超参数，我们通过不断的实验调参，最终确定  $\alpha = 0.25$ ,  $\gamma = 2.0$ . 本文中使用的 Focal Loss 损失函数为：

$$L_{focal}(p, gt) = \begin{cases} -0.25(1-p)^{2.0} \log(p) & \text{if } gt = 1; \\ -0.25(1-p)^{2.0} \log(1-p) & \text{if } gt = 0. \end{cases} \quad (2-10)$$

最后，我们综合 Dice Loss 和 Focal Loss 两个损失函数，将它们相加。因此，本文对 3D-UNet 网络基准模型使用的最终的损失函数为：

$$TotalLoss = L_{dice}(Cuboid_{pred}, Cuboid_{gt}) + L_{focal}(p, gt) \quad (2-11)$$

特别需要指出的是，我们将 Dice Loss 损失与 Focal Loss 损失相加求和，从数值上看可能有相互抵消或增强的意味，但它们对梯度的计算是相互独立的。根据梯度的计算方法，*TotalLoss* 的梯度  $\nabla TotalLoss$  由  $L_{dice}$  和  $L_{focal}$  的梯度  $\nabla L_{dice}$ 、 $\nabla L_{focal}$  分别计算而来，在误差反向传播的过程中，它们都会更新参数的，而不会相互抵消或增强的。

### 2.2.2.3 Adam 优化器与动态调整学习率

支气管气道树分割是属于数据分布稀疏的场景，即有效的支气管体素在整个 CT 体素中的占比非常小，我们可以从 ATM22 数据集中随机挑选 5 个病例，统计支气管体素在整个 CT 体素的占比来看稀疏程度。从表2-1可以看出，支气管体素占比都不

表 2-1 支气管体素占比

Table 2-1 The percent of bronchus voxels over total voxels

病例名称	支气管体素	总体素	支气管体素占比 (%)
ATM_009_0000	115,366	209,453,056	0.055
ATM_027_0000	48,503	161,218,560	0.030
ATM_016_0000	182,904	187,170,816	0.097
ATM_033_0000	127,045	171,704,320	0.073
ATM_023_0000	141,790	209,453,056	0.067

到 0.1%，因此我们弃选随机梯度下降 (Stochastic Gradient Descend, SGD) 优化器。为了更好地利用稀疏梯度信息，达到更好地收敛，我们选择 Adam 优化器。但是对 Adam 优化器，我们做了一些修改，采用动态调整学习率。让我们来看看 Adam 优化器<sup>[52]</sup>计算梯度和更新参数的过程，来解释我们为什么采用动态调整学习率的方法。

令网络模型的参数向量  $\theta$  在  $t$  时间步

$$\theta_t = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}_t \quad (2-12)$$

Adam 优化器会首先计算在  $t$  时间步的梯度  $grad_t$

$$grad_t = \nabla_t J(\theta_t) \quad (2-13)$$

依据梯度的指数移动平均，计算在  $t$  时间步的一阶矩估计  $m_t$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) grad_t \quad (2-14)$$

其中  $\beta_1$  为指数衰减率，控制权重分配、动量和当前梯度，默认取 0.9。

依据梯度平方的指数移动平均，计算在  $t$  时间步的二阶矩估计  $v_t$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) grad_t^2 \quad (2-15)$$

其中  $\beta_2$  为指数衰减率，控制之前梯度的平方的影响，对梯度的平方进行加权平均，默认取 0.999。

由于一阶矩估计  $m_t$  初始化为 0，即  $m_0 = 0$ ，这会导致  $m_t$  偏向于 0，特别是在训练早期阶段。为此我们需要对  $m_t$  进行偏差纠正，降低偏差对训练早期的影响。

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2-16)$$

其中  $\beta_1^t$  表示计算指数衰减率  $\beta_1$  的  $t$  次幂。

同样地，我们也需要对二阶矩阵估计  $v_t$  进行偏差纠正。

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2-17)$$

$\beta_2^t$  类似地表示对指数衰减率  $\beta_2$  的  $t$  次幂。

最后我们来更新在  $t$  时间步的参数  $\theta_t$ ，学习率  $lr$  乘以梯度均值  $\widehat{m}_t$  与梯度方差的平方根  $\sqrt{\widehat{v}_t}$  之比。为防止除零而引入平滑常数  $\epsilon = 1 \times 10^{-8}$

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}_t = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}_{t-1} - lr * \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \quad (2-18)$$

在式2-18中，若分母  $\sqrt{\hat{v}_t} + \epsilon$  过小，就会产生过大的参数  $\theta_t$ ，这会导致在接近最优值的“山峰”或“山谷”时会反复振荡，所以我们采取降低学习率的方法来稍微抵消一些。在训练的过程中，我们会逐渐地动态地降低学习率。具体降低学习率的策略是：如果训练时的损失保持在一个平台超过 10 个 epoch，我们就会降低学习率 10 倍，在训练后期我们会将这个周期拉长为 20 个 epoch。初始学习率设置为  $3 \times 10^{-3}$ ，动态降低学习率按照表 2-2 的规律来进行调整。

表 2-2 动态降低学习率

Table 2-2 Dynamically descent the learning rate

Epoch #	Learning rate
1	$3 \times 10^{-3}$
11	$3 \times 10^{-4}$
21	$3 \times 10^{-5}$
41	$3 \times 10^{-6}$
60	$3 \times 10^{-6}$

### 2.2.3 模型实现和运行环境

本文的模型是采用 Python v3.10.9 版本编写的，基于 PyTorch v1.13.1 框架实现 3D-UNet 模型的。本文的代码编写调试和模型训练均是在上海交通大学高性能计算中心提供的 AI 超算平台上进行的。该 AI 超算平台由 8 台 DGX-2 组成，每台 DGX-2 配备 16 块 NVIDIA Tesla V100 显卡，深度学习张量计算能力达到 16PFLOPS。详细的配置与计算能力如表2-3所示：本文的实验被分配最多 8 块显卡，由于有多个对比实

表 2-3 人工智能超算平台资源

Table 2-3 The computation resource of AI supercomputer platform

队列	每个节点的配置参数	节点数量
dgx2	CPU: 2*Intel Xeon Scalable Cascade Lake 8168 (2.7GHz, 24 cores) Memory: 1.5TB DDR4 ECC REG 2666 GPU: 16*NVIDIA Tesla V100	8 (16 卡/节点)

验同时运行，因此不是 8 块显卡被一个实验独占使用。一般地我为轻计算量的实验分配 2 块显卡，为重计算量的实验分配 4 块显卡。这样在同一个时间段就可以同时执行多个计算任务，使对比实验可以并行进行。

由于本文的实验是在超级计算机上执行，完全不同于传统的普通台式机、工作站等环境，因此有必要讲解清楚本文的实验运行环境。AI 超算平台使用 Slurm 作业调度系统（如图2-10）来执行超算用户提交的计算任务，用户需要编写专有的 Slurm 作业脚本来提交作业请求。Slurm 脚本里指定当前作业需要用到的 CPU 核数，需要用

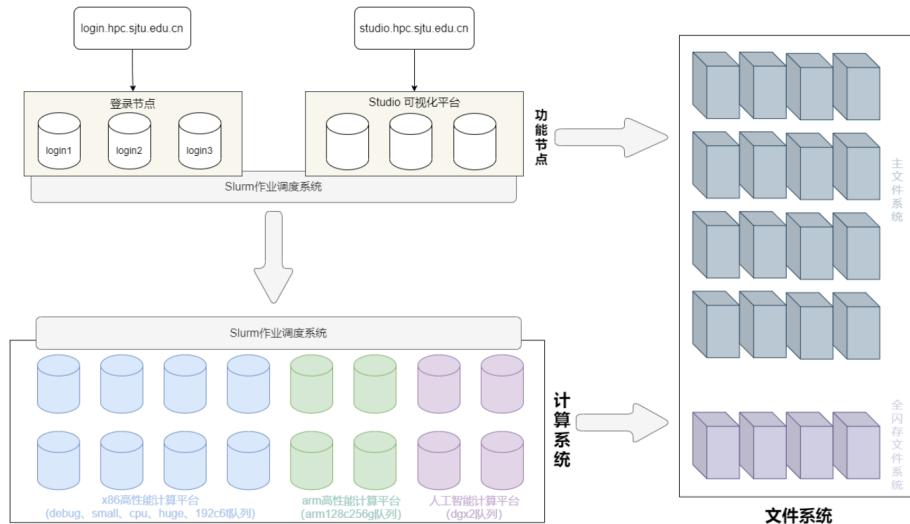


图 2-10 AI 超算平台 Slurm 作业调度系统 [53]

Figure 2-10 The Slurm job scheduling system on AI supercomputer platform

到多少张 GPU 显卡，多少个进程，占用多少个计算节点。这是大规模并行计算需要指定的计算资源。除此之外，还要配置 Python-PyTorch 环境，用户准备好可执行程序启动方式，设置好参数选项。最后通过 \$sbatch Airway3DSegment\_Baseline.slurm 命令提交作业，Slurm 作业调度系统分配好计算资源后才开始执行计算任务。详情请见附录A.3的 Airway3DSegment\_Baseline.slurm 作业脚本代码。

### 2.3 数据集 ATM22

本文使用的数据集是公开的 ATM22 数据集<sup>[38-41]</sup>，是由上海交通大学医疗机器人研究院联合上海胸科医院从多家医疗机构，多台不同品牌型号的 CT 扫描仪收集 500 例就诊者的胸腔扫描图像，然后由三名具有五年以上专业经验的放射科医生对气道树结构进行精细的标注。ATM22 数据集融合了 EXACT'09 数据集和 LIDC-IDRI 部分病例的数据。我们随机选取了 66 例 CT 扫描图像作为训练集，9 例作为验证集，19 例作为测试集。训练集、验证集和测试集的比例基本按照 70:10:20 的比例来。由于 ATM22

数据集是采自不同医疗机构、不同品牌型号的 CT 扫描仪，因此我们对数据进行了预处理，将这些 CT 图像的体素强度统一截断在  $[-1000, 400]$  亨式单位 Hounsfield unit 的窗口范围内，并归一化到  $[0, 1]$  范围。

ATM22 数据集里的 CT 图像切片的尺寸基本都是  $H512 \times W512$  个像素，存在很大面积的黑色背景，为了避免学习到肺部之外的无关区域，我们对其进行裁切，只保留肺的最小包围框作为有效的区域，输入给深度网络进行学习。裁切前后的肺部区域效果对比如图 2-11 所示，裁切后不仅可以减少计算量，最重要的是排除无关区域对

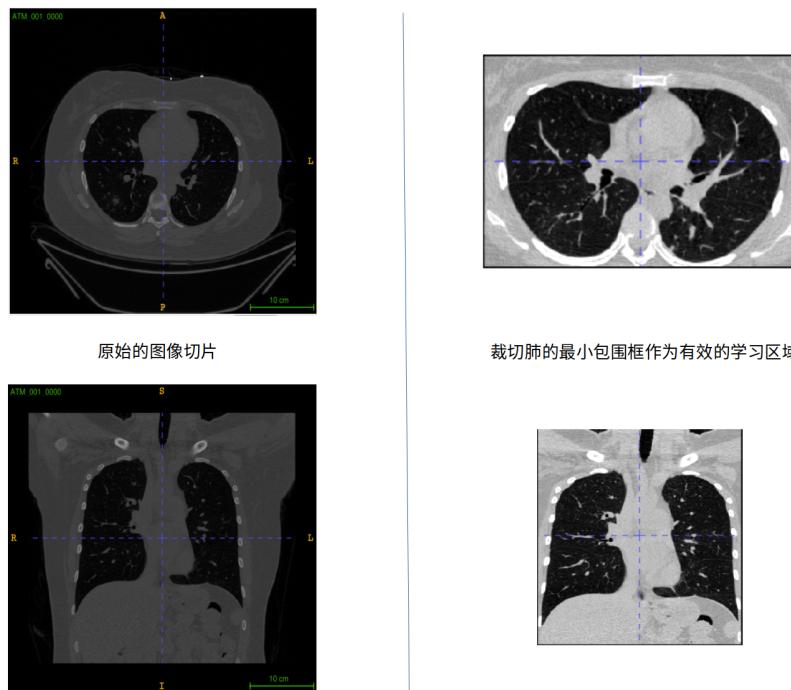


图 2-11 裁切 CT 图像切片的效果对比

Figure 2-11 The pulmonary region comparison before and after cropping CT image slice

网络的权重与偏置参数产生影响。

经过上述的裁切后，CT 图像体的体积大幅缩小，就拿图 2-11 的 ATM\_001\_0000 病例来说，CT 图像体从  $D679 \times H512 \times W512$  缩减到  $D595 \times H225 \times W333$ 。但即使如此，对于单个 GPU 单元显存来说仍然比较大，为此，我们将 CT 图像体切割成  $D80 \times H192 \times W304$  尺寸的长方体子块。本文使用由上海交通大学高性能计算中心提供的 AI 超算平台来进行训练与计算的。AI 超算的显卡型号为 NVIDIA Tesla V100，切割成这样的长方体子块最大化利用了 GPU 的资源又不至于撑爆，可以允许调整 Batch size。本文的网络在训练时分别使用了 1/2/4/8 四种不同的 Batch size，经测试验证

$D80 \times H192 \times W304$  的长方体子块在 GPU 显存中均运行良好。我们对训练集、验证集和测试集的 CT 图像体采用了如此相同的切割方法和尺寸。当然，你可能根据你的 GPU 显存资源决定切割的长方体子块的大小，显存资源比较大则可以选择切割成更大的长方体子块。我们对 CT 图像体采用滑动窗口的切割方式，在训练阶段滑动窗口的大小为  $D64 \times H96 \times W152$ , 在验证和测试阶段则的滑动窗口大小为  $D64 \times H72 \times W72$ . 滑动窗口的意思是每次切割之前沿着宽度 W、高度 H、深度 D 方向逐次移动指定的距离。

为了增强训练效果，也是为了大幅扩大量数据量，我们对每个切割的长方体子块图像块进行了数据增强，具体的数据增强方式有：

1. 沿深度、高度、宽度三个轴随机翻转 RandomFlip
2. 随机仿射变换并重采样 RandomAffine
3. 随机高斯滤镜模糊图像 RandomBlur
4. 给图像添加随机高斯噪声 RandomNoise
5. 随机运动模糊，使图像产生运动伪影 RandomMotion
6. 添加随机 spike 伪影，在图像空间产生不同方向的条纹 RandomSpike
7. 添加随机残影 RandomGhosting

以上数据增强方式，我们通过组合不同的增强方式来联合增强切割的长方体子块图像数据。我们还加入随机概率作用于这些数据增强方式，进一步增强图像数据。

## 2.4 评价指标和分割效果可视化

对于支气管气道树的分割任务，其目标是分割出精确的支气管三维模型，用于临床辅助诊疗。评价支气管气道树分割质量的好坏，有如下指标：

1. 假阳性率 False Positive Rate, FPR

假阳性代表着误检，也就是说将本不是支气管的体素错误地判定为支气管体素。

假阳性率越高表示错误检查发生越多，分割结果愈不可信。

2. 假阴性率 False Negative Rate, FNR

假阴性代表着漏检，就是将本来是真实的支气管体素漏掉了，而认为是普通的背景体素。假阴性过高会导致临床诊断时可能将潜藏的真实疾病漏诊，而贻误了治疗时机。

3. 灵敏度 Sensitivity

灵敏度也叫真阳性率 (True Positive Rate, TPR)，其表示检测出来的真实支气管

体素占实际的全部支气管体素的比例。

#### 4. 精度 Precision

精度表示检测出来的真实支气管体素跟真实的支气管体素与发生误检的假阳性支气管体素之和的比例，精度代表着模型的分割能力。在 95% 的置信区间内，模型能辨识出实际的真实支气管体素的能力。精度越高，表示分割出来的支气管气道树越接近患者的真实情况，就能更可靠地帮助临床医生做出准确的病情诊断。

#### 5. 骰子相似度系数 Dice Similarity Coefficient, DSC

DSC 用来衡量网络分割的结果与金标准之间的相似性，是一种集合相似度度量函数。表示为预测的支气管体素与真实的支气管体素的交集跟预测的支气管体素加上真实的支气管体素进行比较。

#### 6. 检测到的分支 Branch Detected, BD

BD 表示模型检测到的分支数相比于参考的实际存在的分支总数

#### 7. 检测到的树长 Tree Length Detected, TLD

树长为模型检测到的所有分支的中心线长度之和，TLD 表示检测到的树长相比于参考的真实的树长。

上述这些评价指标是在 EXACT'09<sup>[29]</sup>上被提出来的，并被普遍接受和广泛使用的评价指标。评估的重点是放在提取最完整的气道树，进入更高代的气道<sup>①</sup>并提取尽可能多的分支。

如何计算这些评价指标，我们以模型预测的支气管体素对比真实的支气管体素的比对表2-4来说明，并定义计算方法。依据表2-4，上述的指标按如下的公式计算：

##### 1. 假阳性率

$$FPR = \frac{Voxel_{FP}}{Voxel_{FP} + Voxel_{TN}} \times 100\% \quad (2-19)$$

##### 2. 假阴性率

$$FNR = \frac{Voxel_{FN}}{Voxel_{FN} + Voxel_{TP}} \times 100\% \quad (2-20)$$

<sup>①</sup> 注：在医学上通常使用代来表示支气管分支的相对关系。从咽喉部一直到肺部气管第一个分岔前的气管称为第 0 代气管，从第一个分岔出来的左右肺主气管称为第 1 代气管，然后分岔到上/中/下肺叶的气管称为第 2 代气管，再分岔到段支气管的称为第 3 代气管。如此类推，每分岔一个支气管，气管的代就增加 1。气管的代越高，表示气管管腔直径就越细小。

表 2-4 评价指标计算方法说明  
Table 2-4 The explanation of metrics calculation

模型预测的支气管体素	真实的支气管体素	表示	体素个数	在 ITK-SNAP 中 <sup>[54]</sup> 的颜色标记
1	0	假阳性	$Voxel_{FP}$	绿色
1	1	真阳性	$Voxel_{TP}$	蓝色
0	0	真阴性	$Voxel_{TN}$	黑色(背景)
0	1	假阴性	$Voxel_{FN}$	红色

### 3. 灵敏度

$$Sensitivity = \frac{Voxel_{TP}}{Voxel_{TP} + Voxel_{FN}} \times 100\% \quad (2-21)$$

### 4. 精度

$$Precision = \frac{Voxel_{TP}}{Voxel_{TP} + Voxel_{FP}} \times 100\% \quad (2-22)$$

### 5. 骰子相似度系数

$$DSC = \frac{2 * Voxel_{TP}}{(Voxel_{TP} + Voxel_{FN}) + (Voxel_{TP} + Voxel_{FP})} \times 100\% \quad (2-23)$$

### 6. 检测到的分支

$$BD = \frac{Branch_{seg}}{Branch_{gt}} \times 100\% \quad (2-24)$$

其中  $Branch_{seg}$  表示分割结果中检测到的分支数,  $Branch_{gt}$  表示真实存在的分支总数

### 7. 检测到的树长

$$TLD = \frac{Len_{seg}}{Len_{gt}} \times 100\% \quad (2-25)$$

其中  $Len_{seg}$  表示分割结果中检测到的所有分支的中心线长度之和,  $Len_{gt}$  表示真实存在的所有分支的中心线长度之和。

经过模型训练后分割出来的支气管气道树, 我们可以将其导入到 ITK-SNAP<sup>[54]</sup>软件中显示分割效果。这里我们选择 ATM\_054\_0000 这个病例来可视化支气管气道树分割 3D 模型 (这是经过 3D-UNet 基准网络分割的结果), 如图 2-12 所示。图例中绿

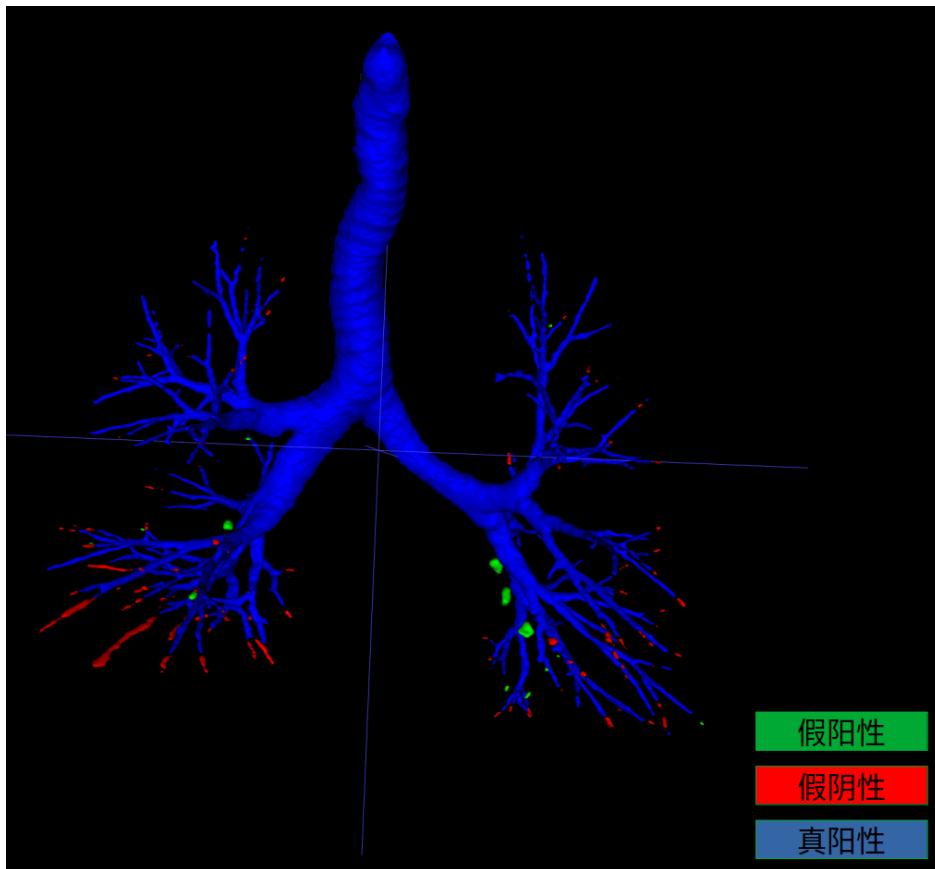


图 2-12 支气管气道树分割的 3D 模型

Figure 2-12 The 3D model of airway tree after segmentation

色的表示假阳性体素，红色的表示假阴性体素，蓝色的表示真阳性体素。

假阳性体素说明分割模型发生了误检，从图2-12可以明显看出绿色的体素跟气道树枝杆是分离的，不属于支气管体素，而应该是黑色的背景体素，但3D-UNet分割模型却错误地认为是支气管体素。假阴性说明分割模型发生了漏检，红色的体素原本是真实存在的支气管，但3D-UNet分割模型漏掉了这些真实存在的支气管体素。这些漏掉的真实存在的支气管体素都是在支气管气道树的末端，是属于非常细小的支气管，3D-UNet分割模型没有把这些细小的支气管体素分割出来，说明分割模型的能力还不够精细。最后，蓝色的体素则是表明分割模型分割出来的支气管体素与真实存在的支气管体素是完全吻合的。粗大的气管、左右肺主支气管、上/中/下肺叶支气管和较细的段支气管都被正确地分割出来，但在更细小的小叶支气管分割能力方面则显得还不够。

需要指出的，本中文对支气管气道树分割的颜色标记统一为绿色表示假阳性，红

色表示假阴性，蓝色表示真阳性。全文保持一致的颜色标记，后文中若没有特别指出颜色标记的意义，均视为与此处一致，不再重复地给出图例说明。

## 2.5 实验结果与分析

### 2.5.1 3D-UNet 基准网络训练过程

我们将 3D-UNet 基准网络放在上海交通大学高性能计算中心的 AI 超级计算机上进行训练，输入是从 ATM22 数据集上随机挑选 66 例肺部 CT 扫描图像，按照 2.3 节介绍的裁切方法去掉无关的黑色背景区域，只保留肺部最小包围框作为有效学习区域。然后将这些经裁切的 Cuboid 体数据按照  $D64 \times H96 \times W152$  的滑动窗口步长随机切割成  $D80 \times H192 \times W304$  的子块，每个子块都经过 2.3 节所述的 7 种数据增强方法进行增强。Label 数据也是 66 例同病例的 CT 体数据，按照同样的方式进行裁切和切割成  $D80 \times H192 \times W304$  的子块，但这些 Label 子块不进行数据增强。每一个图像 Cuboid 体数据子块与对应的 Label 体数据子块绑定在一起。训练集、验证集和测试集的 CT 图像切割情况如表 2-5 所示。我们还会在将这些子块送给网络进行学习前会进

表 2-5 训练集、验证集和测试集的数据一览

Table 2-5 Overview of the cropped CT cube images among trainset, validateset and testset

	CT 扫描图像 例数	切割成 $D80 \times H192 \times W304$ 的子块总数
训练集	66	2202
验证集	9	415
测试集	19	770

行一次洗牌，完全打乱 2202 个子块的顺序，每一个迭代周期都会洗牌一次。同样地，我们也对验证集的 415 个子块和测试集的 770 个子块在每个迭代周期都进行洗牌。

在训练过程中，我们使用 TensorBoard 收集每一个迭代周期 Epoch 的损失函数值，我们密切注视着损失函数的曲线。我们观察到 40 多个迭代周期后，损失函数曲线已经平缓了，趋于收敛了，在 60 个迭代周期后我们便结束了训练过程。训练过程中的损失函数曲线如图 2-13 所示。在训练过程中，第一个迭代周期我们就对验证集的 9 例 CT 图像进行分割，输出每例 CT 图像的支气管气道树 3D 模型和指标数据。当然这个分割结果肯定是非常糟糕的，但我们将它视为检验程序正确运行的参照物，训练迈出第一步的标志。之后每经过 10 个迭代周期，程序就会对验证集和测试集里的 CT 图

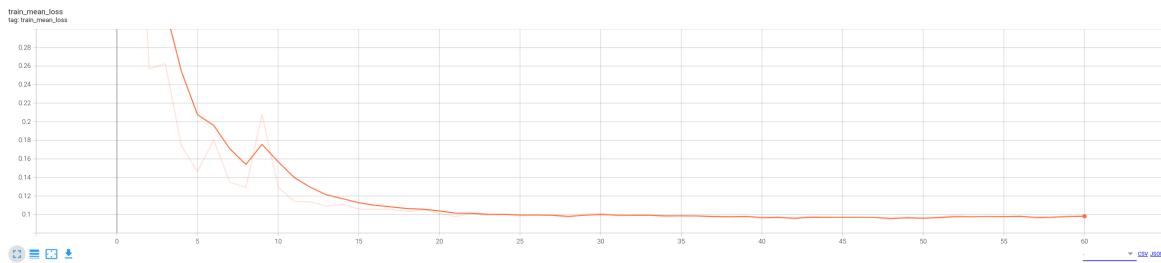


图 2-13 3D-UNet 基准网络训练时的损失函数曲线

Figure 2-13 The loss function curve of 3D-UNet baseline network during training process

像进行分割，输出并保存每例 CT 图像的支气管气道树 3D 模型和指标数据。每 5 个迭代周期都会保存模型的 Check Point 状态和参数，并以 model\_epochnum.ckpt 文件名（如第 35 个迭代周期的模型名字就是 model\_035.ckpt）格式保存下来。这样可使我们能够在训练中断后沿着上次的 model\_xxx.ckpt 继续训练，而不用浪费时间从头开始训练。训练完成后将模型 Check Point 状态和参数保存为 model\_latest.ckpt，作为后续验证过程和测试过程的预训练模型，直接加以使用。除了保存模型 Check Point 状态后参数外，我们在每 10 个迭代周期将指标数据保存进 TensorBoard。累积这些数据帮助我们观察网络的优化发展状况，分析这些数据帮助我们找到改进方向。

在验证过程中，我们直接使用预训练模型 model\_latest.ckpt 对验证集里的 9 例 CT 图像进行分割，输出每例 CT 图像的支气管气道树 3D 模型和指标数据。在测试过程中，我们对测试集里的 19 例 CT 图像执行同样的操作。我们将在 2.5.2 节展示这些实验结果。

以上的训练、验证和测试过程均在一次提交的 Slurm 作业里完成。3D-UNet 基准网络使用了 2 张 NVIDIA Tesla V100 显卡，Batch-size 设置为 4，即一次读入 4 个  $D80 \times H192 \times W304$  的子块，2202 个子块需要 551 次循环才能算完，启动 4 个进程进行并行计算。训练过程每个迭代周期平均耗时 658.7 秒，验证过程跟训练过程同样的设置，每迭代周期平均耗时 4137.6 秒，而测试过程每迭代周期平均耗时 9309.6 秒。详细的运行时间见表 2-6。为什么每迭代周期验证耗时和测试耗时都明显比训练耗时要长得多？这里需要解释一下。验证过程需要对 9 例 CT 图像计算评价指标数据，还需要计算产生三维气道树模型。每计算一个三维气道树模型需要耗时 6 分钟之久，计算 BD 和 TLD 两个指标也需要耗时 5 分钟左右。重要的是，计算三维气道树模型和 BD、TLD 指标数据无法使用 GPU 进行并行计算而加速，只能采用 CPU 串行计算方式。这主要是三维气道树模型和 BD、TLD 指标的计算涉及到很多的逻辑判断，加之

Python Numpy 库函数没有 GPU 版本的，所以无法使用 GPU 加速计算。验证集有 9

表 2-6 3D-UNet 基准网络训练、验证、测试耗时一览表

Table 2-6 Time consumption of training, validating and testing on 3D-UNet baseline

运行时间(单位: 秒)							
迭代周期	训练耗时	验证耗时	测试耗时	迭代周期	训练耗时	验证耗时	测试耗时
1	745.56	5034.34		31	653.84		
2	656.85			32	656.55		
3	657.74			33	655.44		
4	657.58			34	655.85		
5	657.91			35	655.63		
6	657.84			36	655.75		
7	657.86			37	656.22		
8	657.01			38	657.3		
9	656.64			39	656.49		
10	656.53	5068.17	9560.42	40	656.66	3770.93	9457.15
11	663.29			41	654.77		
12	656.7			42	656.61		
13	657.82			43	656.63		
14	658.24			44	656.56		
15	658.56			45	656.46		
16	657.96			46	656.78		
17	658.15			47	656.46		
18	657.97			48	656.14		
19	658.68			49	655.84		
20	658.03	3770.34	9557.97	50	655.97	3770.28	8337.74
21	660.52			51	656.05		
22	656.01			52	655.59		
23	656.75			53	655.69		
24	655.99			54	656.75		
25	656.48			55	657.62		
26	659.39			56	657.26		
27	658.22			57	658.17		
28	659.65			58	658.94		
29	658.81			59	657.99		
30	658.45	3780.8	9544.24	60	658.28	3768.32	9399.98
61		5252.09		62			9726.56
平均	658.7	4137.6	9309.6	总耗时		38 小时 50 分	

例 CT 图像，测试集有 19 例 CT 图像都需要计算指标数据和产生三维支气管气道树模型，所以每个迭代周期测试集计算耗时是验证集计算耗时 2 倍多。但好在我们不是每个迭代周期都会去计算验证集和测试集，而是每隔 10 个迭代周期才去计算一次。验证集在第一个迭代周期被计算一次，而测试集并没有被计算。

表2-6反映了我们的整个实验过程。经历了 60 个迭代周期训练过程完成。在

第 61 个迭代周期，我们使用训练获得的最新模型参数（即加载最新的预训练模型 model\_latest.ckpt）来执行验证过程，第 62 个迭代周期执行测试过程。最后，3D-UNet 基准网络训练、验证和测试全过程总耗时 38 小时 50 分钟，完成一个完整实验。**需要说明的是：此处的总耗时 38 个多小时是训练、验证和测试进行 62 个迭代周期所耗费的时间。在实际应用中，当部署好预训练模型后，输入一个肺部 CT 扫描图像到输出支气管气道树分割结果的时间只需 10 秒左右。这个分割速度已经能够满足临床术前的要求。**

## 2.5.2 实验结果和分析

上述的训练、验证和测试过程完成后，我们来展示实验结果。由于篇幅的原因，我们挑选验证集里的 9 例 CT 图像，展示支气管气道树 3D 模型的可视化效果和他们的指标数据。而对于测试集里的 19 例 CT 图像，19 张支气管气道树 3D 模型图片实在太多放不下，我们只展示 19 组指标数据。

我们将进行横向和纵向的对比。

- A. 在横向对比时，我们对验证集的 9 例 CT 图像，每例 CT 图像均匀地取 5 张切片，选取中间的那张切片。排列这 9 张切片看看它们的分割效果和指标数据，如表2-7所示。

需要说明的是，表2-7中的指标数据是基于当前切片计算的，不是基于整个气道树来计算的，因此就不带有 BD 和 TLD 两个指标的数据了。还有一点，切片图像中由绿色像素框起来的区域是真实的气管，红色的像素是分割出来的。绿色像素覆盖在红色像素上，对于没有覆盖住而露出来的红色像素是假阳性的。

- B. 在纵向对比时，我们挑选 ATM\_054\_0000 这个病例，将第 10 个迭代周期，第 20/30/40/50/60 个迭代周期里的 Slice #264 拿出来对比，看看它们的分割效果和指标数据，如表2-8所示。

从表2-7看，ATM\_174\_0000 Slice #74 的 Sensitivity 和 DSC 两项指标表现得最差，这是因为该 CT 图像的切片数量最少，其支气管气道明显狭窄，可能患有肺部疾病。倒数第二差的是 ATM\_505\_0000 Slice #177 这张切片，存在很多红色的像素表明假阴性像素很多，它们通常是一些比较细小的支气管。分割网络遗漏掉了这些真实存在的支气管像素。在这 9 例 CT 切片图像中分割最好的当属于 ATM\_688\_0000 Slice #292, Sensitivity, Precision, DSC 分别取得了 97.62%, 94.411%, 96.15% 的高分。这表明分割网络对于肺叶支气管、段支气管这种中等管腔直径的支气管能很清晰地分割出来。

表 2-7 验证集 CT 切片图像分割效果与指标横向比较

Table 2-7 Horizontal comparison of segmentation and metrics in the validate set

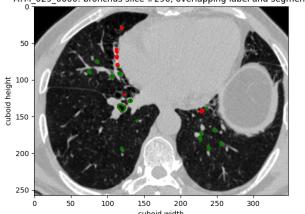
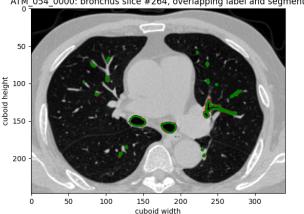
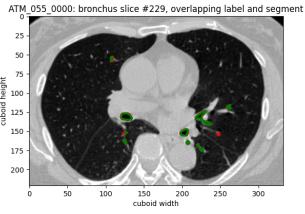
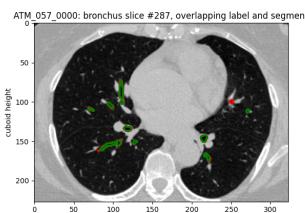
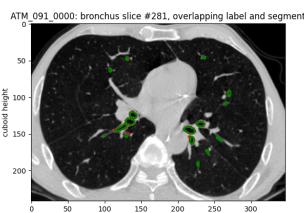
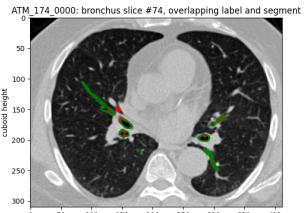
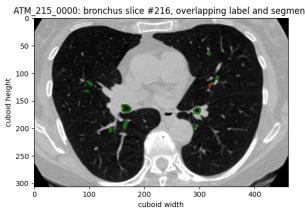
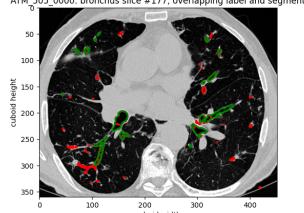
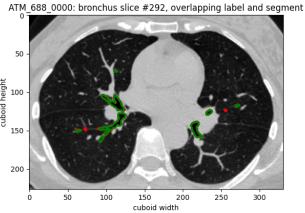
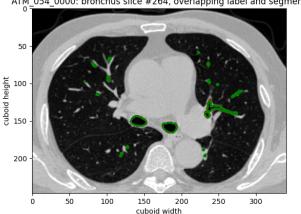
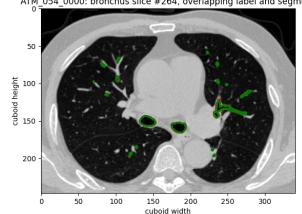
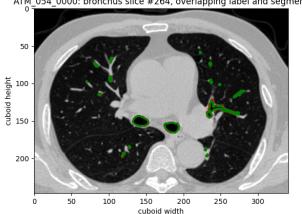
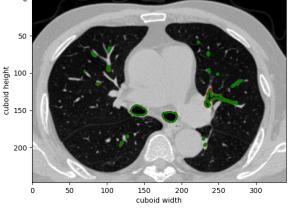
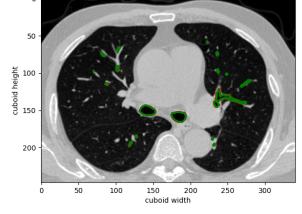
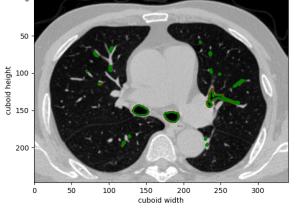
<b>ATM_029_0000</b> Slice #296  FPR = 0.066% FNR = 14.286% Sensitivity = 85.714% Precision = 72.558% DSC = 78.59%	<b>ATM_054_0000</b> Slice #264  FPR = 0.068% FNR = 5.991% Sensitivity = 94.009% Precision = 92.039% DSC = 93.01%	<b>ATM_055_0000</b> Slice #229  FPR = 0.056% FNR = 9.966% Sensitivity = 90.034% Precision = 86.469% DSC = 88.22%
<b>ATM_057_0000</b> Slice #287  FPR = 0.089% FNR = 5.06% Sensitivity = 94.94% Precision = 83.073% DSC = 88.61%	<b>ATM_091_0000</b> Slice #281  FPR = 0.068% FNR = 3.419% Sensitivity = 96.581% Precision = 88.802% DSC = 92.53%	<b>ATM_174_0000</b> Slice #74  FPR = 0.044% FNR = 69.159% Sensitivity = 30.841% Precision = 82.5% DSC = 44.9%
<b>ATM_215_0000</b> Slice #216  FPR = 0.021% FNR = 18.069% Sensitivity = 81.931% Precision = 90.068% DSC = 85.81%	<b>ATM_505_0000</b> Slice #177  FPR = 0.026% FNR = 44.246% Sensitivity = 55.754% Precision = 72.892% DSC = 63.18%	<b>ATM_688_0000</b> Slice #292  FPR = 0.05% FNR = 2.038% Sensitivity = 97.962% Precision = 94.411% DSC = 96.15%

表 2-8 验证集 ATM\_054\_0000 病例第 264 张切片图像分割效果与指标纵向比较

Table 2-8 Vertical comparison of segmentation and metrics for ATM\_054\_0000 slice #264

ATM_054_0000 Epoch #10	ATM_054_0000 Epoch #20	ATM_054_0000 Epoch #30
 <p>FPR = 0.025% FNR = 12.268% Sensitivity = 87.732% Precision = 96.698% DSC = 92.0%</p>	 <p>FPR = 0.061% FNR = 6.419% Sensitivity = 93.581% Precision = 92.786% DSC = 93.18%</p>	 <p>FPR = 0.066% FNR = 6.419% Sensitivity = 93.581% Precision = 92.264% DSC = 92.92%</p>
ATM_054_0000 Epoch #40	ATM_054_0000 Epoch #50	ATM_054_0000 Epoch #60
 <p>FPR = 0.066% FNR = 6.277% Sensitivity = 93.723% Precision = 92.275% DSC = 92.99%</p>	 <p>FPR = 0.068% FNR = 6.277% Sensitivity = 93.723% Precision = 92.017% DSC = 92.86%</p>	 <p>FPR = 0.068% FNR = 5.991% Sensitivity = 94.009% Precision = 92.039% DSC = 93.01%</p>

从表2-8看，随着训练的持续进行，从第 20 个迭代周期开始，分割性能就逐渐开始稳定下来。Sensitivity, Precision 和 DSC 都已经达到了 92% 以上的优秀表现。不管是对于左右肺主气管、肺叶支气管，还是很细小的段支气管都可能清晰地分割出来，证明 3D-UNet 基准网络性能基本上能达到临床应用的水平。

最后我们来看看 3D-UNet 网络对整个支气管气道树的分割表现。对于测试集的 19 例 CT 图像的分割，其性能指标如表 2-9所示。其中黑色加粗打双下划线的表示在当前指标表现最出色的。从表2-9可以看出，ATM\_245\_0000 这一例 CT 图像分割表现最好，BD 和 TLD 两个指标竟然达到例 100%，这说明已经将支气管气道树中的分

表 2-9 测试集上 3D-UNet 基准网络对支气管气道树分割的性能指标表现

Table 2-9 The metrics table of 3D-UNet baseline model segmenting airway tree in testset

病例名称	FPR	FNR	Sensitivity	Precision	DSC	BD	TLD
ATM_001_0000	<b><u>0.006</u></b>	7.24	92.76	<b><u>98.089</u></b>	95.35	68.95	84.17
ATM_024_0000	0.031	7.236	92.764	90.916	91.83	84.18	92.86
ATM_034_0000	0.022	3.705	96.295	96.251	96.27	90.75	93.93
ATM_041_0000	0.047	4.808	95.192	87.189	91.02	82.38	90.38
ATM_060_0000	0.024	2.534	97.466	93.12	95.24	88.03	92.72
ATM_061_0000	0.028	3.121	96.879	91.897	94.32	86.15	90.8
ATM_074_0000	0.053	3.838	96.162	87.038	91.37	80.49	89.86
ATM_075_0000	0.023	4.633	95.367	93.65	94.5	81.13	88.56
ATM_080_0000	0.026	4.067	95.933	93.183	94.54	76.9	88.11
ATM_150_0000	0.082	3.902	96.098	80.036	87.33	72.12	87.62
ATM_158_0000	0.044	3.461	96.539	86.953	91.5	80.75	90.48
ATM_163_0000	0.035	4.301	95.699	91.958	93.79	83.23	91.34
ATM_197_0000	0.032	9.234	90.766	90.81	90.79	59.71	78.46
ATM_245_0000	0.038	<b><u>0.309</u></b>	<b><u>99.691</u></b>	83.496	90.88	<b><u>100</u></b>	<b><u>100</u></b>
ATM_246_0000	0.04	0.611	99.389	82.853	90.37	<b><u>100</u></b>	<b><u>100</u></b>
ATM_260_0000	0.016	1.041	98.959	94.227	<b><u>96.53</u></b>	98.34	97.93
ATM_266_0000	0.019	2.351	97.649	93.112	95.33	99.38	98.39
ATM_271_0000	0.049	1.044	98.956	86.544	92.33	97.6	97.62
ATM_638_0000	0.014	3.152	96.848	95.199	96.02	90.48	95.45

支全部检测出来，也获得了最长的分支中心线长度之和。

对于验证集的 9 例 CT 图像，我们不仅要展示支气管气道树分割的可视化结果，如图2-14所示。

还有完整的指标数据，见表2-10。在图2-14中我们看到 ATM\_174\_0000 和 ATM\_505\_0000 两个病例中出现大量的红色的假阴性体素。结合表2-10中这两个病例的 BD、TLD 两项指标低至 30% 的不正常现象<sup>①</sup>，而 Precision 能够达到 90% 以上。纵览这 9 张气道树 3D 分割图，其他 7 张图都没有在气管、左右肺主气管发生漏检。所以，我们推断 ATM\_174\_0000 和 ATM\_505\_0000 两个病例的支气管气道树异常的原因可能是：

- a) 图像异位，做标注工作的临床医生没有看到明确的支气管管壁边缘而没有标注；
- b) 患者有严重的肺部疾病，支气管管壁破裂或缺失。

<sup>①</sup> 表2-10中红色标记的数据

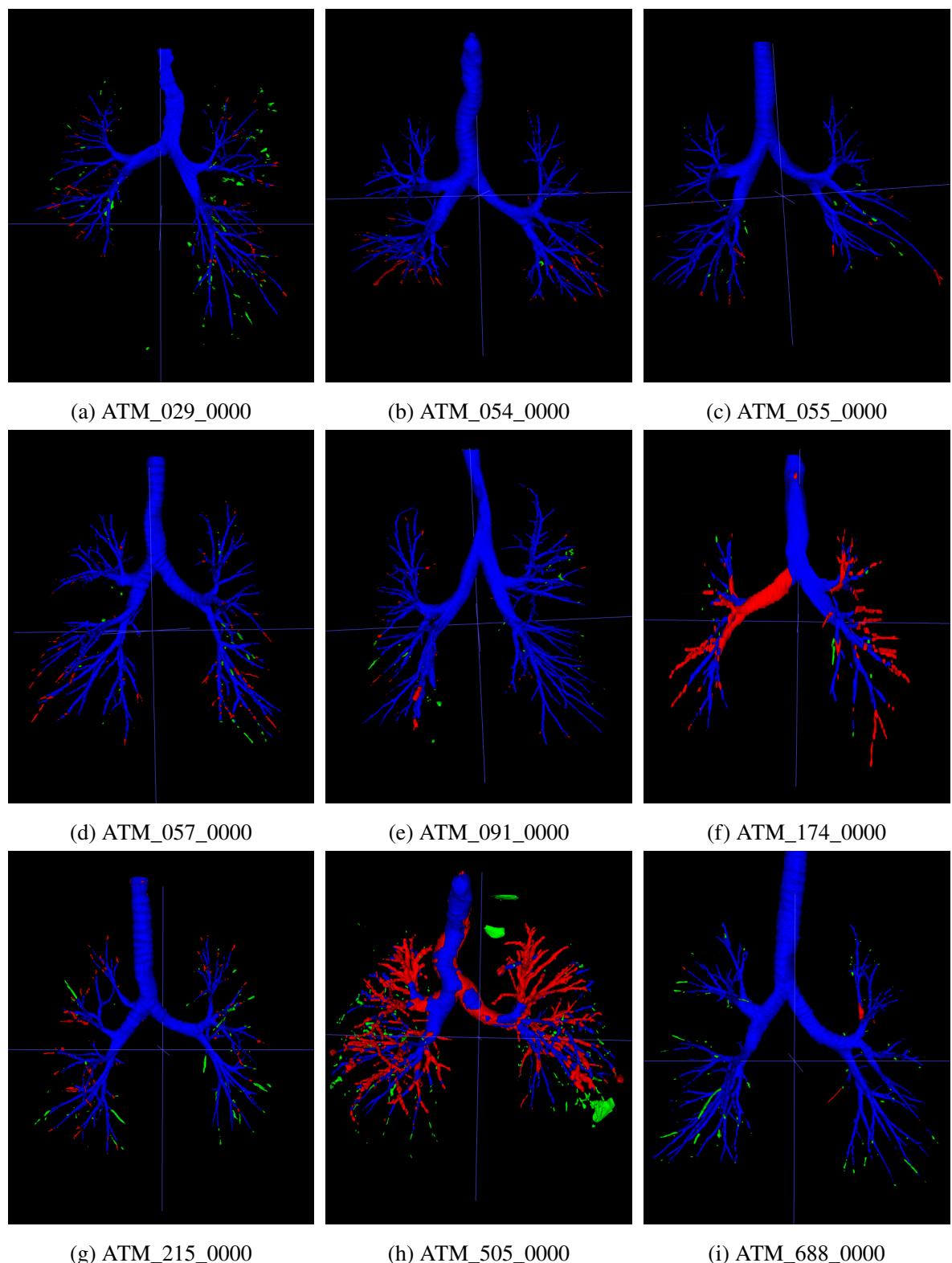


图 2-14 验证集 9 例 CT 图像的支气管气道树分割可视化 3D 模型

Figure 2-14 Visualize the airway 3D model of 9 CT cases in validate set

表 2-10 验证集 9 例 CT 图像的支气管气道树分割指标  
Table 2-10 The segmentation metrics of 9 CT cases in validate set

病例名称	FPR	FNR	Sensitivity	Precision	DSC	BD	TLD
ATM_029_0000	<b>0.017</b>	9.222	90.778	93.232	91.99	80.34	88.48
ATM_054_0000	0.032	3.876	96.124	93.343	94.71	76.78	85.61
ATM_055_0000	0.031	3.571	96.429	91.66	93.98	82.51	89.9
ATM_057_0000	0.03	5.757	94.243	92.35	93.29	72.85	84.36
ATM_091_0000	0.034	3.432	96.568	91.148	93.78	84.04	91.09
ATM_174_0000	0.019	26.864	73.136	<b>94.026</b>	82.28	<b>32.38</b>	<b>47.19</b>
ATM_215_0000	0.023	9.419	90.581	91.951	91.26	73.3	84.83
ATM_505_0000	0.05	55.899	44.101	84.808	58.03	<b>21.5</b>	<b>39.1</b>
ATM_688_0000	0.023	<b>2.217</b>	<b>97.783</b>	93.409	<b>95.55</b>	<b>93.07</b>	<b>95.73</b>

我们更倾向于后者。

在此需要强调一点，我们是将 3D-UNet 作为我们的基准网络，所以在此没有做消融研究 Ablation Study，没有凸显 3D-UNet 网络比其他已知网络更优秀。从上述的实验结果展示和分析，我们认识到 3D-UNet 网络存在不足之处，后面的章节以 3D-UNet 网络为基础，在此之上进行改进。

## 2.6 本章小结

本章详细讲述了 3D-UNet 这一基准网络，我们在基于 CT 扫描图像的支气管气道树分割任务中采用 3D-UNet 网络来对肺部支气管进行分割，我们使用 Python 语言与 PyTorch 深度学习框架设计实现了一个 3D-UNet 分割网络，针对 ATM22 数据集进行特定的优化设计。我们在上海交通大学高性能计算中心的 AI 超算上训练网络，训练完成后在验证集和测试集上输出完整的支气管气道树三维模型和评价指标数据。我们展示了实验结果，可视化显示支气管气道树的 3D 模型，分析这些实验数据，并对两个明显异常的病例进行分析推断异常的产生原因。

通过以上实验证明了 3D-UNet 网络可以胜任三维支气管气道树的分割任务，但在精度准确率、FPR 和 DSC 等指标方面还不够，需要进一步改进 3D-UNet 网络。后文在此实验的基础上研究如何改进该网络，以更好更精确地分割支气管气道树三维模型。



### 第三章 注意力蒸馏方法对基准模型的改进

我们在第二章讨论了 3D-UNet 网络这一基准模型对支气管气道树的分割，仔细观察表?? 中各个支气管气道树末端（ATM\_174\_0000 和 ATM\_505\_0000 两个病例除外）出现很多红色的假阴性支气管体素，见图3-1 中放大的左右末梢支气管。假阴性表示这些体素原本是真实存在的支气管体素，只是因为我们的 3D-UNet 基准网络的分割能力不够，尚无足够的精细分割能力。末梢支气管的管径通常都比较小，做标注

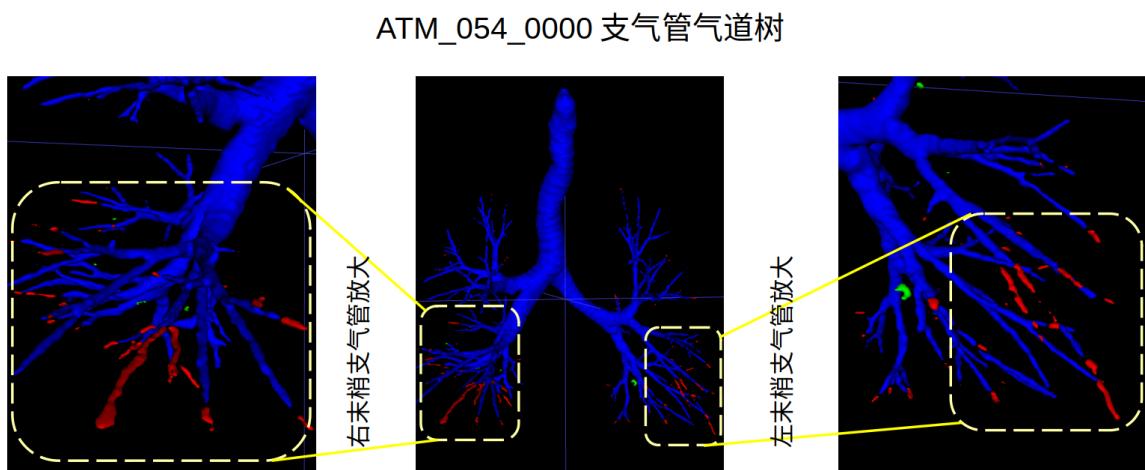


图 3-1 末梢支气管出现假阴性分割效果

Figure 3-1 The distal bronchus segmentation showed false-negative voxels

工作的临床医生在标注这些末梢支气管尽很大的努力，可能只能看到一到两个像素。ATM\_054\_0000 切片上的像素间距是 0.83mm，切片间距是 0.5mm，由此可见末端支气管的管径在人体肺部实际不到 1mm。对于医生来说，精细标注确实是一个比较大的挑战，大量 CT 图片的精细标注是一项繁重耗时且枯燥乏味的工作。但对于支气管镜导航手术来说，越是进入末梢支气管越能接近肺癌结节部位，越好做到微创甚至无创取样活检。这就要求我们做支气管气道树分割越精细越好，较高要求是能分割到段支气管、小叶支气管。当然最高要求是做到能分割细支气管，细支气管是最末端的毛细支气管，直接连到肺泡。

对于卷积网络提取精细特征的要求，Sergey Zagoruyko 等人<sup>[55]</sup>提出卷积网络注意力转移 Attention Transfer 的方法，注意力转移将知识从教师网络转移到学生网络，可以改善 CNN 的分割性能。Geoffrey Hinton 和 Jeff Dean 等人<sup>[56]</sup> 提出知识蒸馏的概念，

指出注意力图是一种有价值的知识。受此启发，知识可迁移，注意力帮助聚焦，我们提出注意力蒸馏的方法。在学习支气管气道树的特征时，加强对末梢支气管这些细小对象的关注，可帮助提高 3D-UNet 网络的分割性能。

### 3.1 注意力蒸馏方法

注意力是一组空间地图，本质是试图使网络在做出输出决策时最关注的输入空间区域<sup>[55]</sup>。有两种注意力图的形式，一种是基于激活的注意力图 Activation-based attention map，另一种是基于梯度的注意力图 Gradient-based attention map。这种注意力图可在网络的各个层定义，以便它们能够捕获低级、中级和高级别的表示信息。注意力图可指导我们看哪里，把目光聚焦在感兴趣的地方。

在卷积网络的第  $n$  层，定义注意力图  $AttMap_n$  作用在该层特征  $Feat_n$  的函数关系：

$$\mathcal{F} : Feat_n^{C \times D \times H \times W} \longrightarrow AttMap_n^{1 \times D \times H \times W} \quad (3-1)$$

这里上标  $D \times H \times W$  表示维数，因为 3D-UNet 是以三维长方体子块体数据为输入的，就像 UNet 以二维的图像数据为输入的。二维图像数据的维度是  $Height \times Width$ ，三维长方体子块体数据的维度是  $Depth \times Height \times Width$ 。而  $C$  表示通道维数，也就是在第  $n$  层卷积有  $C$  个  $D \times H \times W$  个长方体子块。

如何计算 3-1 式的函数关系？Sergey Zagoruyko 等人在论文 *Paying more attention to Attention: Improving the performance of convolutional neural networks via Attention Transfer* 中提出了三种计算方式<sup>[55]</sup>：

1. 沿着通道维数，计算绝对值之和

$$\mathcal{F}_{sum}(Feat_n) = \sum_{c=1}^C |Feat_n| \quad (3-2)$$

2. 沿着通道维数，计算绝对值的  $p$  次幂之和

$$\mathcal{F}_{sum}^p(Feat_n) = \sum_{c=1}^C |Feat_n|^p \quad (3-3)$$

3. 沿着通道维数，计算绝对值的  $p$  次幂的最大值

$$\mathcal{F}_{max}^p(Feat_n) = \max_{c=1}^C |Feat_n|^p \quad (3-4)$$

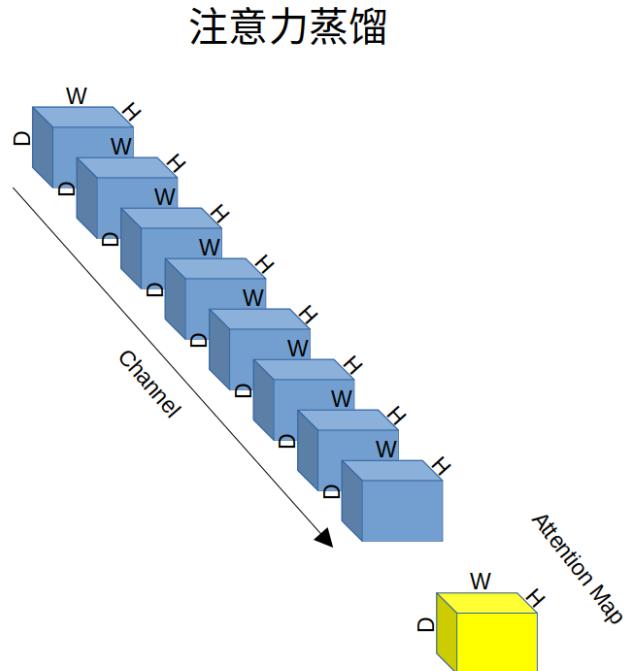
在这里我们选择第 2 种计算方式，沿着通道维数，计算绝对值的  $p$  次幂之和。因为这种计算方式保留了该层特征的所有隐含的显著激活信息，但不会忽略非最大的元素，也不会削弱判别性的元素。 $p$  次幂建议取  $p = 2$ ，它增强了大多数敏感任务区域。 $\mathcal{F}_{sum}^p(Feat_n)$  的计算相比于  $\mathcal{F}_{sum}(Feat_n)$ ，它把更多的权重放置于最具判别性的部位。 $p$  越大，越能聚焦于这些判别性部位。

借助于 PyTorch 的张量表示方法，我们将注意力图函数改写成如下形式：

$$AttMap_n = \mathcal{F}_{sum}^p(Feat_n) = \sum_{c=1}^C |Feat[c, :, :, :]|^p \quad (3-5)$$

其中  $Feat[c, :, :, :]$  表示通道维  $c$ 、深度维  $D$  全部索引、高度维  $H$  全部索引、宽度维  $W$  全部索引，后面的  $[:, :, :]$  就构成了一个  $D \times H \times W$  的长方体子块。

这种计算注意力图的方式其意义就是将  $C$  个  $D \times H \times W$  长方体子块的特征信息蒸馏浓缩在一个  $D \times H \times W$  长方体子块特征里。如图3-2所示：



沿着通道维  $C$ ，累积  $n$  个 Cuboid (蓝色的)，  
蒸馏浓缩到 1 个 Cuboid (黄色的)。

图 3-2 注意力蒸馏的原理

Figure 3-2 How the attention distillation works

为了保证蒸馏得到的注意力图 Cuboid 与原来的特征 Cuboid 保持相同的  $D \times H \times W$  维数，我们需要对注意力图进行三线性插值 Trilinear Interpolation  $\mathcal{L}(\bullet)$

$$AttMap_n = \mathcal{L} [\mathcal{F}_{sum}^p (Feat_n)] \quad (3-6)$$

最后，我们使用 *Softmax* 对其归一化，使每个体素的值被限定在  $[0, 1]$  之间。

$$AttMap_n = Softmax\{\mathcal{L} [\mathcal{F}_{sum}^p (Feat_n)]\} \quad (3-7)$$

注意力图的引入给卷积网络的层与层之间相当于增加了一个额外的梯度，使相邻两层之间更接近，可以使  $AttMap_n$  与  $AttMap_{n+1}$  之差最小化，即它们的累积损失最小。

$$Loss = \min \sum_{n=1}^{N-1} \left\| AttMap_n - AttMap_{n+1} \right\|_F^2 \quad (3-8)$$

其中  $\|\bullet\|_F$  是指矩阵的 Frobenius 范数。

以上是注意力图的计算过程。引入注意力蒸馏的目的是为了增加对细颗粒度的物体的关注，在本文中就是为了增加对末梢支气管这些细小的支气管体素的关注，从而提高分割的性能。

## 3.2 注意力蒸馏的效果可视化

引入注意力蒸馏方法后，我们可以来看看在卷积网络不同层之间，经过蒸馏后特征的表现如何？通过可视化的方式来查看支气管气道树的焦点变化情况。我们选取了测试集的 ATM\_074\_0000 这个病例，抽取 3D-UNet 结构图2–8上采样路径的第 6、第 7、第 8、第 9 个卷积层的特征信息，将它们显示出来，如图3–3所示。从图3–3可以看出，高亮之处也就是聚焦点沿着 conv6->conv7->conv8->conv9 的方向从粗壮的气管逐渐转移至更细小的末梢支气管，这也是注意力机制在引导我们聚焦看哪里。为了更清晰，我们将注意力蒸馏后的图片转成灰度图并放大，请见图3–4，对比 4 个卷积层蒸馏后焦点转移情况。说明一点，为了便于读者复现此注意力蒸馏后的效果，显示如图3–4的图片，请参考附录A.1的代码片段。

注意力蒸馏方法可使焦点转移，聚焦于看更细节的对象，帮助模型提高精细度的性能。为此，我们对前文的 3D-UNet 基准网络进行改进，引入注意力蒸馏方法。

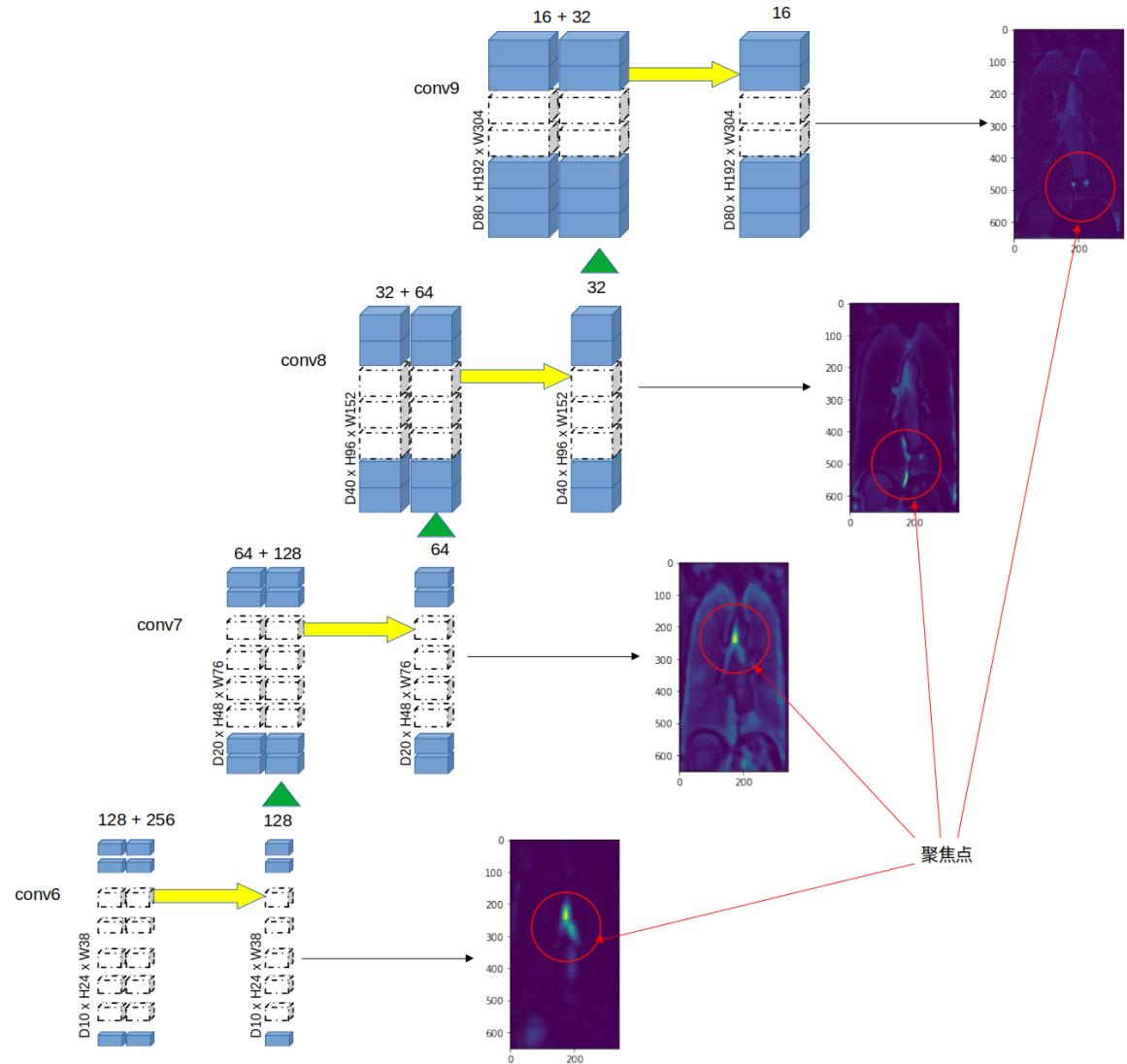


图 3-3 注意力蒸馏后的效果

Figure 3-3 The visual effect of feature after attention distillation

### 3.3 引入注意力蒸馏模块对 3D-UNet 基准网络进行改进

基于前文对注意力蒸馏方法的研究，我们决定对 3D-UNet 基准网络进行改进，引入注意力蒸馏方法，在 3D-UNet 上采样路径上对第 6 个、第 7 个、第 8 个和第 9 个卷积层添加注意力蒸馏模块。经过注意力蒸馏模块后，相邻的蒸馏后图像特征之间就自然增加了一个新的梯度，我们通过 3-8 式来计算它们之间的损失。

改进后的网络结构如图 3-5 所示。因插图横向宽度太长，将其横向放置，利于看清图中细小的字体。我们只是在上采样路径上加入了注意力蒸馏模块，而并没有在下

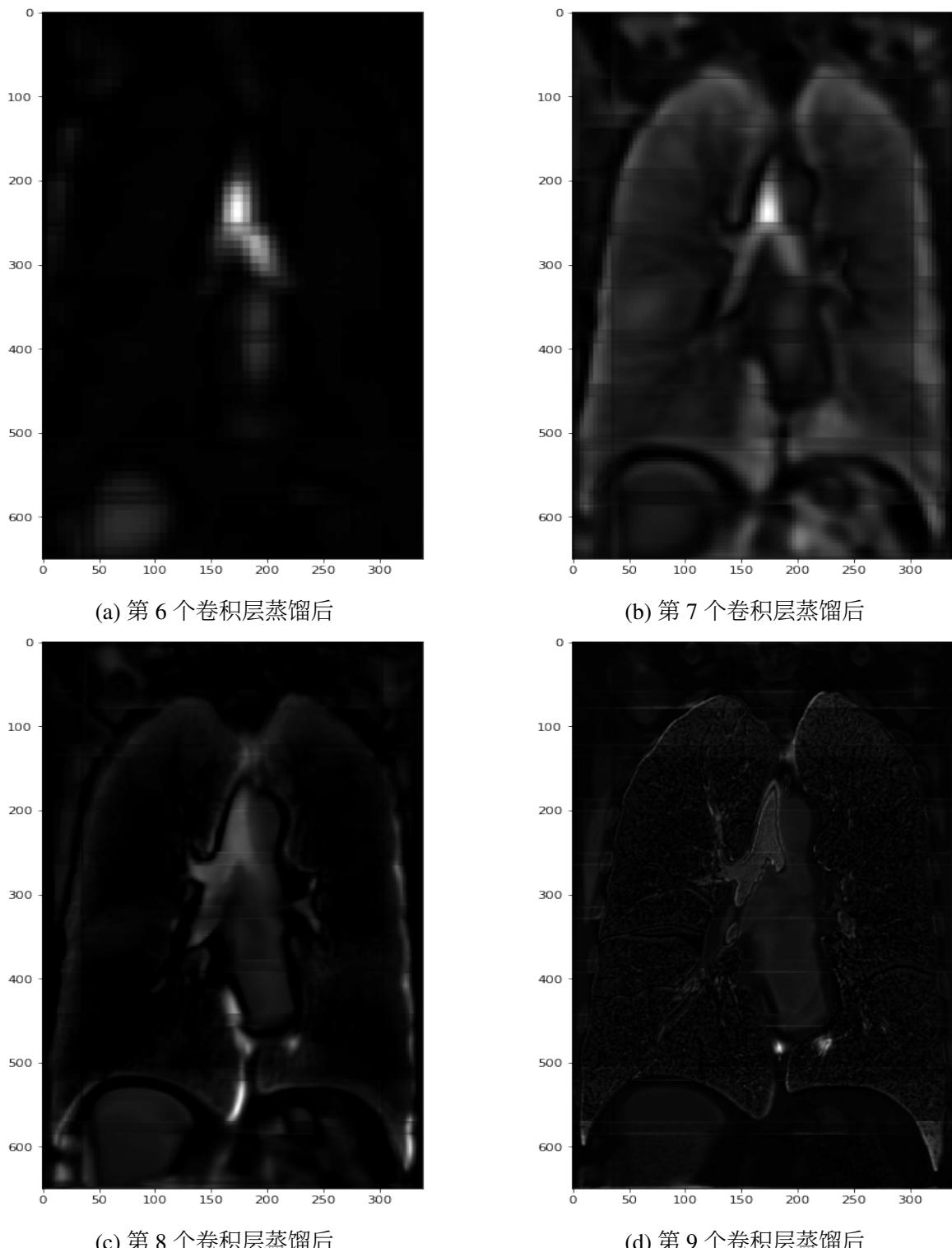


图 3-4 注意力蒸馏后图像特征对比

Figure 3-4 The comparison of image feature after attention distillation

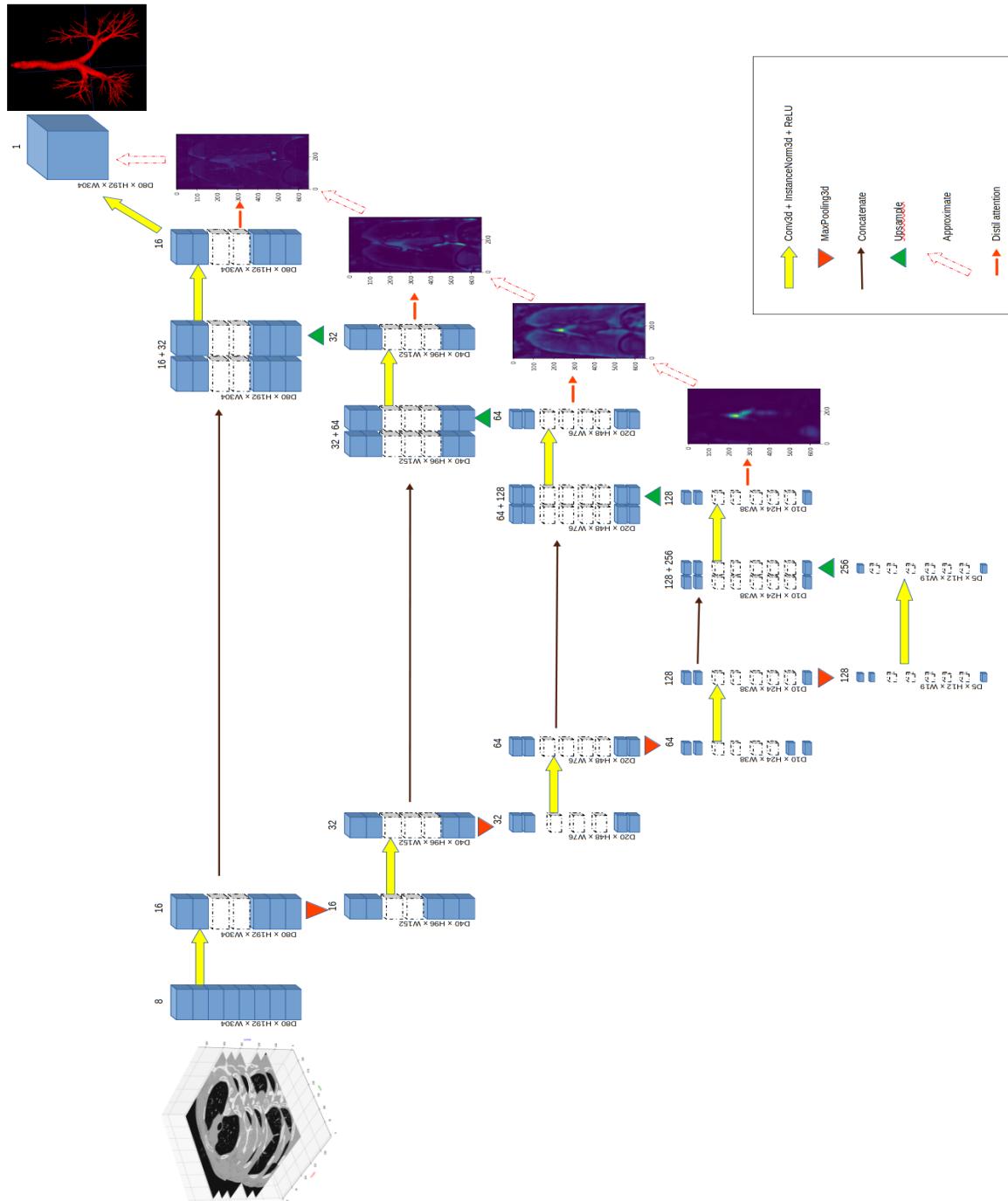


图 3-5 3D-UNet + Attention Distillation 网络结构示意图

Figure 3-5 The architecture of 3D-UNet + Attention Distillation

采样路径上同样加入注意力蒸馏模块，为什么？这是考虑到下采样路径上第  $n$  个卷积层本身就比第  $n - 1$  个卷积层的输出图像的分辨率要低，这就必然导致末梢支气管的

像素密度更低。在低分辨率下，注意力蒸馏就很难聚焦于末梢支气管，所以就难以起到正向促进作用。而在上采样路径则正好相反，第  $n$  个卷积层比第  $n - 1$  个卷积层的图像分辨率提高一倍，末梢支气管的像素密度高，那么注意力蒸馏就可以比较容易地聚焦于这些末梢支气管。

对于注意力蒸馏模块引入后额外增加的梯度，其带来的损失函数的计算，实现算法 3-1，需要累积各个卷积层的注意力蒸馏的梯度损失。骰子损失 Dice Loss 还是按

---

### 算法 3-1 注意力蒸馏的梯度损失

---

```

Input: AttMap[4], γ[3]
Result: Total_Loss
1 Total_Loss ← 0
2 γ[3] ← [0.1, 0.1, 0.1]
3 for  $i \leftarrow 0$  to length(AttMap) - 1 do
    // 将第  $i$  个 AttMap 值赋给预测值
    pred ← AttMap[i]
    // 将第  $i+1$  个 AttMap 值赋给再 groundtruth
    groundtruth ← AttMap[i + 1]
    // 对预测值进行三线性插值
    pred ← torch.nn.functional.interpolate(pred, mode='trilinear',
                                           scale_factor=2)
    N = pred.size(0)
    // 对预测值和真实值分别进行展平
    flatten_pred ← pred.view(N, -1)
    flatten_gt ← groundtruth.view(N, -1)
    // 预测值和真实值分别进行 Softmax 归一化
    pred_softmax ← torch.nn.functional.softmax(flatten_pred, dim=1)
    groundtruth_softmax ← torch.nn.functional.softmax(flatten_gt, dim=1)
    // 计算它们的均方差，并作为当前循环的损失
    loss ← torch.nn.functional.mse_loss(pred_softmax, groundtruth_softmax)
    // 累积每次循环的损失，合并到总的损失
    Total_Loss ← Total_Loss + γ[i] * loss
14 end

```

---

照2-4式进行计算，Focal Loss 损失按照公式2-10计算，最后需要将此计算出来的总的注意力蒸馏的梯度损失缀加到 Total Loss（见公式2-11）上。需要指出的是，注意力

蒸馏的梯度损失函数是可微可导的，因此在误差反向传播时是可以更新 3D-UNet 网络的参数的。

### 3.4 对比实验与实验结果分析

我们对 3D-UNet 基准网络进行改进，加入了注意力蒸馏模块后，现在我们进行一次实验，与前文的 3D-UNet 基准网络实验进行对比。使用的数据集与 3D-UNet 实验的数据集完全相同，实验条件也是一致的。本次实验的训练、验证和测试三个阶段合计总耗时约 39 个小时。

验证集的指标数据如表3-1所示。另外算出这些指标的平均值，以便与 3D-UNet

表 3-1 3D-UNet + AD 网络结构的验证集指标数据一览表

表中每一项指标栏用双下划线加粗黑体标示出表现最优秀的，用波浪线标示出表现最差的。

Table 3-1 The metrics overview of validate set under 3D-UNet + AD network structure

病例名称	FPR	FNR	Sensitivity	Precision	DSC	BD	TLD
ATM_029_0000	0.022	9.266	90.734	91.012	<u><u>90.87</u></u>	76.92	85.85
ATM_054_0000	0.037	3.499	96.501	92.319	94.36	76.16	<u><u>83.93</u></u>
ATM_055_0000	0.039	3.453	96.547	89.855	93.08	81.17	89.09
ATM_057_0000	0.039	4.683	95.317	90.426	92.81	78.35	87.98
ATM_091_0000	<u><u>0.041</u></u>	3.57	96.43	<u><u>89.583</u></u>	92.88	85.46	91.99
ATM_215_0000	<b>0.021</b>	<u><u>10.595</u></u>	<u><u>89.405</u></u>	<b><u><u>92.453</u></u></b>	90.9	<u><u>72.25</u></u>	84.32
ATM_688_0000	0.028	<b>2.109</b>	<b>97.891</b>	92.182	<b><u><u>94.95</u></u></b>	<b><u><u>93.94</u></u></b>	<b><u><u>96.31</u></u></b>
平均值	0.032	5.311	94.689	91.119	92.836	80.607	88.496

基准网络实验的验证集的指标的平均值进行比较。

测试集的指标数据如表3-2所示。从平均值来看，FPR、Sensitivity、DSC 和 TLD 四项指标表现得都比较优秀。

我们还可从平均值来比较 3D-UNet 跟 3D-UNet + AD 的指标性能，见表3-3。通过对比，我们看到在 BD, TLD 和 FNR 三项指标上有一些提升，在其他项指标上则基本持平。

还有一个重要的实验结果就是支气管气道树分割效果可视化，我们分别从测试集和验证中挑选一些比较有代表性的病例来展示它们的分割效果，如图 3-6所示。

从图3-6的支气管气道树三维模型来看，大部分的假阴性（红色体素）已经被正确分割出来，9个病例中的假阴性红色体素很少出现了。相比于图2-14里大量出现的

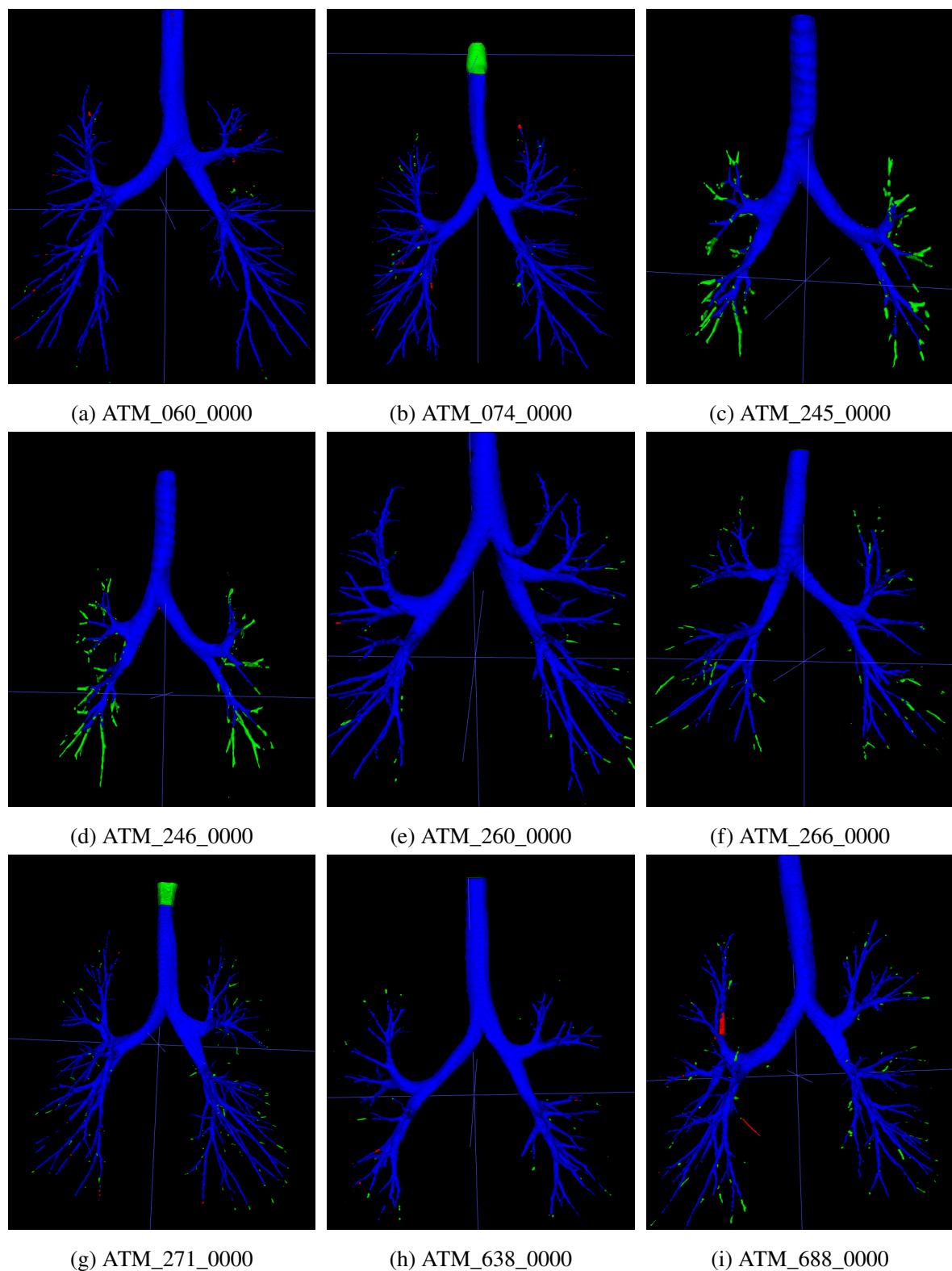


图 3-6 3D-UNet + AD 网络结构支气管气道树分割可视化 3D 模型  
9 个样本分别从测试集和验证集挑选的

Figure 3-6 Visualize the airway tree 3D models under the 3D-UNet + AD network

The 9 samples are selected from test set and validate set.

表 3–2 3D-UNet + AD 网络结构的测试集指标数据一览表  
表中每一项指标栏用双下划线加粗黑体标示出表现最优秀的, 用波浪线标示出表现最差的.

Table 3–2 The metrics overview of testset under 3D-UNet + AD network structure

病例名称	FPR	FNR	Sensitivity	Precison	DSC	BD	TLD
ATM_001_0000	<b>0.007</b>	7.258	92.742	<b>97.779</b>	95.19	71.05	85.1
ATM_024_0000	0.036	6.387	93.613	89.696	91.61	87.54	93.5
ATM_034_0000	0.021	4.039	95.961	<b>96.455</b>	<b>96.21</b>	93.06	94.67
ATM_041_0000	0.055	4.486	95.514	85.247	90.09	83.42	90.69
ATM_060_0000	0.027	2.682	97.318	92.505	94.85	90.17	94.34
ATM_061_0000	0.034	3.152	96.848	90.471	93.55	85.05	90.35
ATM_074_0000	0.06	2.934	97.066	85.671	91.01	91.64	94.44
ATM_075_0000	0.027	3.944	96.056	92.649	94.32	88.68	92.16
ATM_080_0000	0.031	3.662	96.338	92.216	94.23	78.55	88.01
ATM_150_0000	<u>0.085</u>	3.853	96.147	<u>79.511</u>	<u>87.04</u>	74.78	88.6
ATM_158_0000	0.047	3.003	96.997	86.234	91.3	87.87	92.67
ATM_163_0000	0.039	3.726	96.274	91.247	93.69	88.06	94.41
ATM_197_0000	0.032	<u>10.332</u>	<u>89.668</u>	90.723	90.19	<u>64.03</u>	<u>81.64</u>
ATM_245_0000	0.036	<b>0.568</b>	<b>99.432</b>	84.172	91.17	<b>100</b>	<b>100</b>
ATM_246_0000	0.041	0.629	99.371	82.469	90.13	<b>100</b>	<b>100</b>
ATM_260_0000	0.019	1.144	98.856	93.23	95.96	98.9	98.71
ATM_266_0000	0.025	2.077	97.923	91.103	94.39	<b>100</b>	99.24
ATM_271_0000	0.055	1.238	98.762	85.151	91.45	98.4	98.34
ATM_638_0000	0.016	3.52	96.48	94.514	95.49	90.48	95.91
平均值	0.036	3.612	96.388	89.529	92.73	87.983	93.304

表 3–3 3D-UNet 与 3D-UNet + AD 的性能指标对比  
Table 3–3 The metrics comparison between 3D-UNet and 3D-UNet + AD

	FPR	FNR	Sensitivity	Precison	DSC	BD	TLD
验证集							
3D-UNet	<b>0.029</b>	13.362	86.638	<b>91.770</b>	88.319	68.53	78.477
3D-UNet + AD	0.032	<b>5.311</b>	<b>94.689</b>	91.119	<b>92.836</b>	<b>80.607</b>	<b>88.496</b>
测试集							
3D-UNet	<b>0.033</b>	<b>3.715</b>	96.285	<b>90.343</b>	<b>93.122</b>	85.293	92.036
3D-UNet + AD	0.036	3.612	<b>96.388</b>	89.529	92.73	<b>87.983</b>	<b>93.304</b>

红色假阴性体素明显减少了很多（ATM\_174\_0000 和 ATM\_505\_0000 两个严重肺部疾病的病例除外），证明注意力蒸馏的方法抑制住了假阴性率，提高了分割模型的一些性能。

ATM\_688\_0000 这个病例稍微有点例外，但仔细研究是在上叶支气管处出现的红色假阴性体素，上叶支气管的管径是有点粗的，分割模型应该不会在此处发生漏检，最有可能的是医生只是标记了管内壁，没有标记管外壁。分割模型识别到管外壁，在三维支气管气道树上表现为管外壁的红色体素覆盖了管内壁的蓝色体素。在 ITK-SNAP 里放大查看，如图3-7, 证实了支气管管内壁的体素被标注了，而支气管管

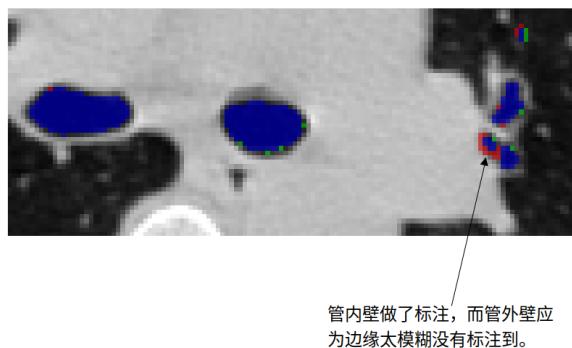


图 3-7 支气管内外壁标注差异而导致覆盖问题

Figure 3-7 The red FN voxels outside the bronchial wall overlapped the blue TP voxels inside the bronchial wall

外壁因为边缘太模糊没有标注，这就出现了管外壁的红色假阴性体素覆盖了管内壁蓝色真阳性体素。

特别需要指出的是，ATM\_074\_0000 和 ATM\_271\_0000 两个病例在气管最上端有非常明显的绿色假阳性体素，其实这并不是分割模型判断错误了。通过查看原始 CT 图像 3-8，确认这段绿色假阳性体素确实是咽喉处的气管。它们是真实存在的气管，但为什么会发生分割模型误判了这些真实存在的气管，原因就是医生遗漏了标注咽喉部位的气管。从这一角度来说，3D-UNet + AD 网络分割模型其实并没有错误判断，反而帮助提高了真阳性率，分割出更多真实的支气管气道树。

### 3.5 本章小结

本章是在 3D-UNet 网络基准模型的基础上进行改进，提出了注意力蒸馏的新方法来提高基准模型对精细的末梢支气管的分割能力。注意力蒸馏方法是受注意力迁

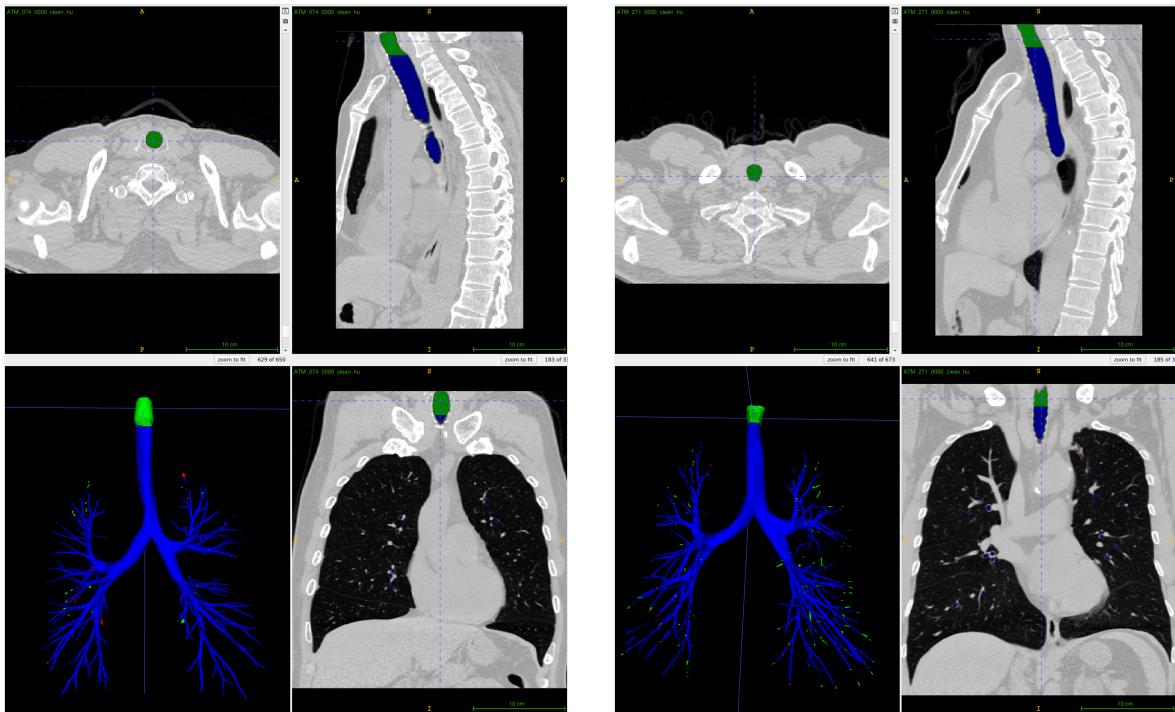


图 3-8 ATM\_074\_0000 和 ATM\_271\_0000 两个特殊病例的分割结果分析

Figure 3-8 The segmentation results analysis for 2 special cases: ATM\_074\_0000 and ATM\_271\_0000

移<sup>[55]</sup>和知识蒸馏<sup>[56]</sup>的启发而创造出来的。我们详细阐述了注意力蒸馏方法的基本原理、计算过程，并给出了注意力蒸馏增加的梯度损失的算法表述。然后我们对注意力蒸馏方法的效果进行可视化展示，验证了注意力（聚焦）从粗大的识别对象逐渐转移到细小的末梢支气管对象。至此我们开始将注意力蒸馏模块引入到 3D-UNet 网路基准模型上，在上采样路径为第 6、第 7、第 8 和第 9 个卷积层添加注意力蒸馏模块，以此改进 3D-UNet 网络。

实现了改进后的网络后，我们进行了对比实验。采用与 3D-UNet 网络基准模型同样的数据集和实验条件，完成训练、验证和测试全过程。从评价指标来看，3D-UNet + AD 网络比 3D-UNet 基准网络取得了进步。最后我们展示了 3D-UNet + AD 网络对支气管气道树分割的可视化效果，可视化效果证实确实抑制了红色假阴性率，提高了对末梢支气管的识别能力。我们还分析了 2 个特殊病例 ATM\_074\_0000 和 ATM\_271\_0000，指出它们的绿色假阳性体素确实是真实存在的气管，模型没有发生误判，还帮助提高了真阳性率。

我们深入观察支气管气道树的三维分割效果，发现在末梢支气管还是存在很多绿色的假阳性体素，这些绿色假阳性体素跟蓝色的真阳性体素其实是断裂开的。那么它

们是否是真实的更纤细的支气管，比如说小叶支气管呢？结合评价指标数据表3-1和表3-2来看，FNR 指标仍然是比较高的，如何进一步降低 FNR 指标？我们将在后文进一步改进现在的 3D-UNet + AD 模型。

## 第四章 通道级特征再学习方法提取精细支气管特征

在前文 3D-UNet 网络基准模型分割出的支气管气道树的基础上, 仔细观察表??和表??里的支气管三维气道树, 一个普遍的规律是: 在气管 Trachea (第 0 代)、左右主支气管 Left/Right main bronchi (第 1 代)、上/中/下叶支气管 Superior/Middle/Inferior lobar bronchus (第 2 代) 这三代支气管上分割良好, 但在段支气管 Segmental bronchi (第 3 代) 分割上出现很多绿色假阳性或红色假阴性体素, 而到了小叶支气管 Lobular bronchi (第 4 代) 的分割上出现了大量的断裂, 最后的细支气管 Bronchiole (第 5 代, 连接到最终的肺泡) 因为太小而完全没有标注, 所以在分割的三维气道树上不存在。对于采用了同样的卷积层和池化层来说, 第 0 代/第 1 代/第 2 代的支气管分割效果明显比第 3 代/第 4 代的支气管要好得多。考虑到 3D-UNet 网络是将切割成  $D90 \times H192 \times W304$  大小的长方体数据作为最小输入单元的。就以标注数据来说, 相同体积的切割体 1 与 2 (见图 4-1) 在不同的空间位置, 它们所容纳的支气管体素密度差异巨大。这个最小

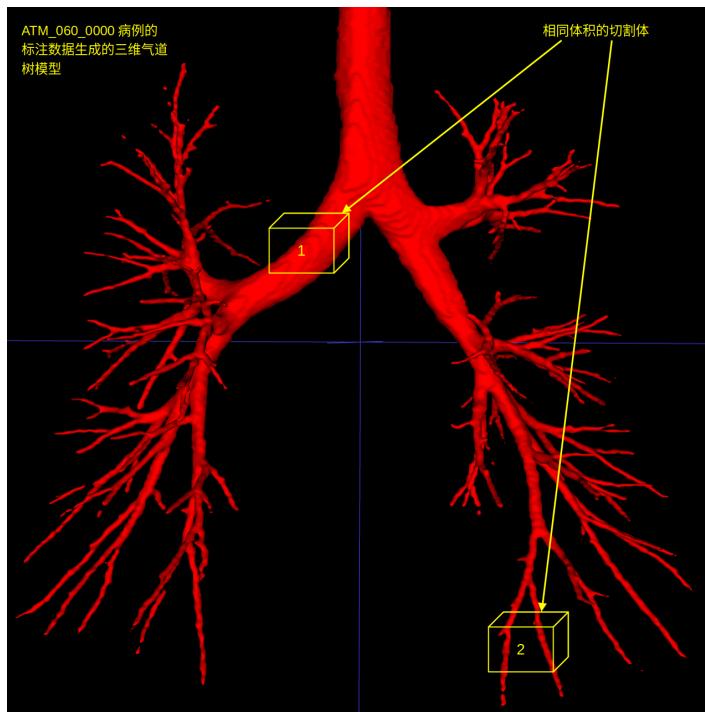


图 4-1 相同体积的切割体在不同空间位置获得的体素密度差异

Figure 4-1 The voxel density difference under different spatial location

输入单元经过卷积层和池化层, 空间分辨率会降低一半。若采用平均池化的方法, 切

割体 2 处的精细特征可能会被“擦除掉”，而切割体 1 处的粗糙特征还能保留下来。即使采用最大池化方法，经过多层卷积和池化后，精细特征也会逐渐消失。因此，我们认为在当前特征和分辨率尺度内，不同的空间位置有不同的重要性。尤其像支气管这样的树状结构，要想获得精细的高代支气管特征，显然需要提高它们的重要性，也就是它们的权重。为了解决这个问题，我们提出一种通道级特征再学习（Channel-wise feature re-learning）方法，拟在卷积层块之后、最大池化层（见图2-6）之前添加一个通道级特征再学习模块来进行再校准，捕获更精细的特征。

## 4.1 通道级特征再学习方法

下面来说明通道级特征再学习方法的基本原理。

给定一个肺部支气管 CT 扫描图像体  $\mathbb{X}_{airway}$ ，则我们的支气管气道树分割任务可描述为式4-1的函数关系，

$$\text{Prob}_{airway} = \mathcal{F}(\mathbb{X}_{airway}) \quad (4-1)$$

s.t.

$$\min \left| \text{Prob}_{airway} - \text{GT}_{airway} \right| \quad (4-2)$$

其中  $\text{Prob}_{airway}$  表示气道的预测概率。我们做支气管气道树分割的目标是通过 3D-UNet 网络学习到一个端到端的映射函数  $\mathcal{F}$ ，使得预测值  $\text{Prob}_{airway}$  与真实值  $\text{GT}_{airway}$  之差最小化。

3D-UNet 网络的最小输入单元是一个三维(立方)体数据，在第  $m$  个卷积层块的特征输出  $\text{Feat}_m$  表示为一个  $C_m \times D_m \times H_m \times W_m$  大小的四维矩阵  $\text{Matrix}_{C_m \times D_m \times H_m \times W_m}$ ，即表示有  $C_m$  个大小为  $D_m \times H_m \times W_m$  的立方体。当  $m = 1$  时，即第一个卷积层块输出的特征含有 16 个大小为  $D80 \times H192 \times W304$  的立方体，这个立方体就是图4-1的一个切割体。如图4-2所示。其中， $C_m$ 、 $D_m$ 、 $H_m$  和  $W_m$  分别表示通道维 (Channel)、深度维 (Depth)、高度维 (Height) 和宽度维 (Width)。借用 PyTorch 的张量表示法，则  $\text{Feat}_m$  中的任意元素

$$\text{Feat}_m[c, d, h, w] \in \text{Matrix}_{C_m \times D_m \times H_m \times W_m} \quad (4-3)$$

其中  $c, d, h, w$  分别指的是通道维、深度维、高度维、宽度维的索引号。

我们要构建的特征再学习就是要定义一个对当前卷积层特征  $\text{Feat}_m$  的通道算子

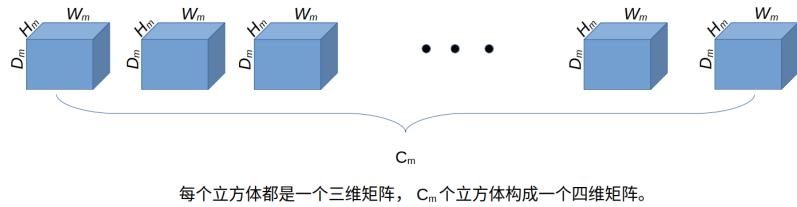


图 4-2 卷积层块的特征的维数

Figure 4-2 The dimensions of feature output from convolutional block layer

$\Psi$ 。那么对应的特征再学习 (Feature Re-Learning, FRL) 可定义为函数关系：

$$FRL_m = \Psi (\text{Feat}_m) \quad (4-4)$$

特征再学习的通道算子是为了校准卷积层特征输出，它是一个通道级的权重图，不仅不仅可以识别特征  $\text{Feat}_m$  的关键空间位置的信息，譬如图4-2 切割体 1 所在位置的粗大枝干的粗糙特征和切割体 2 所在位置的末梢枝节的精细特征，还可以加强影响输出决策的基本通道。我们采用沿着每个空间维度（深度维、高度维、宽度维）的特征加权（通道）组合的方式来集成空间位置的信息。这是一种空间积分的方法，也称为体积分，如4-5式对第  $m$  个卷积层输出特征  $\text{Feat}_m$  的体积分  $\int_V \text{Feat}_m dV$ 。

$$\begin{aligned} \Psi (\text{Feat}_m) &= \int_V \text{Feat}_m dV \\ &= \int_V \text{Feat}_m d(\text{Depth}) \\ &\quad + \int_V \text{Feat}_m d(\text{Height}) \\ &\quad + \int_V \text{Feat}_m d(\text{Width}) \end{aligned} \quad (4-5)$$

我们把这种体积分拆成分别沿深度、高度和宽度 3 个轴方向的积分之和。为了阐述这种体积分是如何进行的，我们从图4-2取出一个立方体来对照着讲解。如图4-3所示，特征立方体的大小为  $D \times H \times W^{\circledR}$ ，每一个小立方体的大小是  $1 \times 1 \times 1$ ，表示为一个体素。

① 本文在描述特征的维数或者张量的 Shape 时，统一按照 PyTorch 对张量的维数从高到低依次为：批次维 Batch、通道维 Channel、深度维 Depth、高度维 Height、宽度维 Width。简写取首字母 [B, C, D, H, W]，全文都采取这样的方式描述。

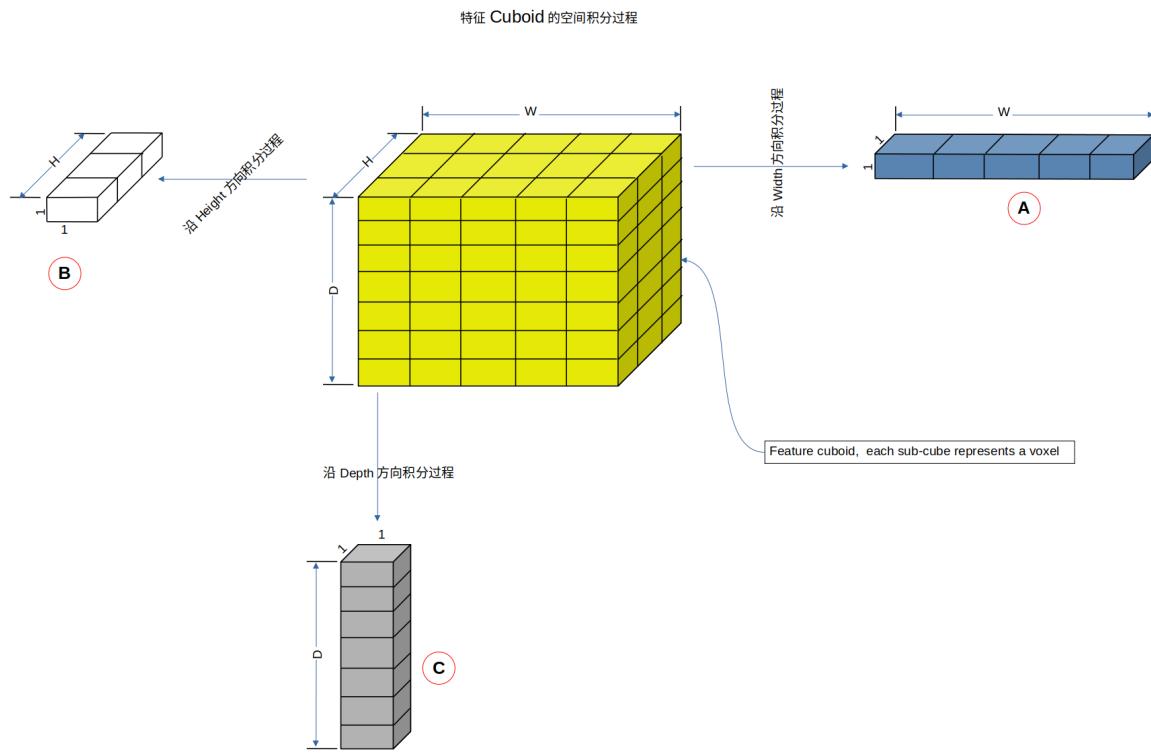


图 4-3 特征立方体的空间积分过程示意图

Figure 4-3 The spatial integration process for feature cuboid

沿  $Width$  方向的积分  $\int_V \text{Feat}_m d(Width)$  过程实际上将  $D \times H \times W$  的特征立方体挤压成  $1 \times 1 \times W$  的形状（图4-3中 A 所表示的形状）。由于在编程实现中特征立方体是一个离散的数据，因此积分转变为一个累积求和过程。沿  $Width$  方向的积分过程是一个两次累积求和过程，借用 PyTorch 的张量表示法，其可表示为式4-6

$$S_1 = \sum_{h=1}^{H_m} \text{Feat}_m[:, h, :] \\ S_2 = \sum_{d=1}^{D_m} S_1[d, :]$$

得到

$$\int_V \text{Feat}_m d(Width) = S_2 = \sum_{d=1}^{D_m} \sum_{h=1}^{H_m} \text{Feat}_m[d, h, :] \quad (4-6)$$

同样的道理，沿  $Height$  方向的积分  $\int_V \text{Feat}_m d(Height)$ ，将特征立方体挤压成  $1 \times H \times 1$

的形状（图4-3中B所表示的形状）。

$$S_1 = \sum_{d=1}^{D_m} \text{Feat}_m[d, :, :] \\ S_2 = \sum_{w=1}^{W_m} S_1[:, w]$$

得到

$$\int_V \text{Feat}_m d(Height) = S_2 = \sum_{w=1}^{W_m} \sum_{d=1}^{D_m} \text{Feat}_m[d, :, w] \quad (4-7)$$

沿 *Depth* 沿方向的积分  $\int_V \text{Feat}_m d(Depth)$ , 将特征立方体挤压成  $D \times 1 \times 1$  的形状（图4-3中C所表示的形状）。

$$\int_V \text{Feat}_m d(Depth) = \sum_{h=1}^{H_m} \sum_{w=1}^{W_m} \text{Feat}_m[:, h, w] \quad (4-8)$$

这种挤压（两次累积求和）的方法借鉴了注意力蒸馏（图3-2）的思想，注意力蒸馏只是沿着通道 *Channel* 方向进行累积求和，而我们在深度 *Depth*、高度 *Height* 和宽度 *Width* 三个方向都进行累积求和。

完成特征的3个方向的积分后，我们将它们加起来，

$$\int_V \text{Feat}_m dV = \sum_{d=1}^{D_m} \sum_{h=1}^{H_m} \text{Feat}_m[d, h, :] \\ + \sum_{w=1}^{W_m} \sum_{d=1}^{D_m} \text{Feat}_m[d, :, w] \\ + \sum_{h=1}^{H_m} \sum_{w=1}^{W_m} \text{Feat}_m[:, h, w] \quad (4-9)$$

我们设计了2个卷积层，如代码片段4.1所示，分别是将原来的通道  $C_m$  降低到  $\frac{C_m}{r}$ ，然后将通道又恢复到  $C_m$ ，这样做的目的是为了重新组合所有的通道。

Listing 4.1 Design 2 conv. layers to recombine the feature channels

```

1 import torch.nn as nn
2
3 DownChannel = nn.Conv3d(in_channels=Cm,
4                         out_channels=Cm//r,
```

```

5             kernel_size=1,
6             stride=1)
7
8 UpChannel = nn.Conv3d(in_channels=Cm//r,
9                     out_channels=Cm,
10                    kernel_size=1,
11                    stride=1)

```

通过这种通道的减少和增加，将多个通道进行重组。从而有信息性的通道得到加强，而无信息性的通道就被抑制削弱。

所以得到的输出为

$$\Psi(\text{Feat}_m) = \text{Sigmoid} \left[ Ku * \text{ReLU} \left( Kd * \int_V \text{Feat}_m dV \right) \right] \quad (4-10)$$

其中  $Kd$  表示 DownChannel 的卷积层,  $Ku$  表示 UpChannel 的卷积层,  $\text{ReLU}$  和  $\text{Sigmoid}$  表示不同的激活函数。

最后，将  $\Psi(\text{Feat}_m)$  与原来的特征  $\text{Feat}_m$  进行逐元素相乘，得到最终的特征输出  $\widehat{\text{Feat}}_m$ 。

$$\widehat{\text{Feat}}_m = \Psi(\text{Feat}_m) \cdot \text{Feat}_m \quad (4-11)$$

以上过程就是通道级特征再学习的原理和计算方法。

卷积层块将特征输出给通道级别特征再学习模块，经过计算后，再输出给最大池化层。在训练过程中，重要的气道（粗糙的枝干支气管和精细的末梢支气管）随着特征再学习模块赋予的新权重的增加逐渐被重视，而无关的区域则随着权重的降低而慢慢被忽略。这就是特征再学习模块所起的作用。

## 4.2 引入通道级特征再学习模块对 3D-UNet 基准网络进行改进

基于前文对通道级特征再学习方法的研究，我们决定对 3D-UNet 基准网络进行改进。引入特征再学习模块，将 3D-UNet 网络的每一个卷积层块后添加一个特征再学习模块。经过特征再学习模块然后传输给最大池化层。修改后整个网络结果如图4-4所示，由于图片的横向宽度太长，明显比纵向高度要大，为了看清图片中的细小字体，将网络结构图横向放置。图4-4中的绿色柱状体表示特征再学习模块产生的新特征。需要说明的是，特征再学习模块并没有改变特征的维数，依然保持与同级别的卷积层块

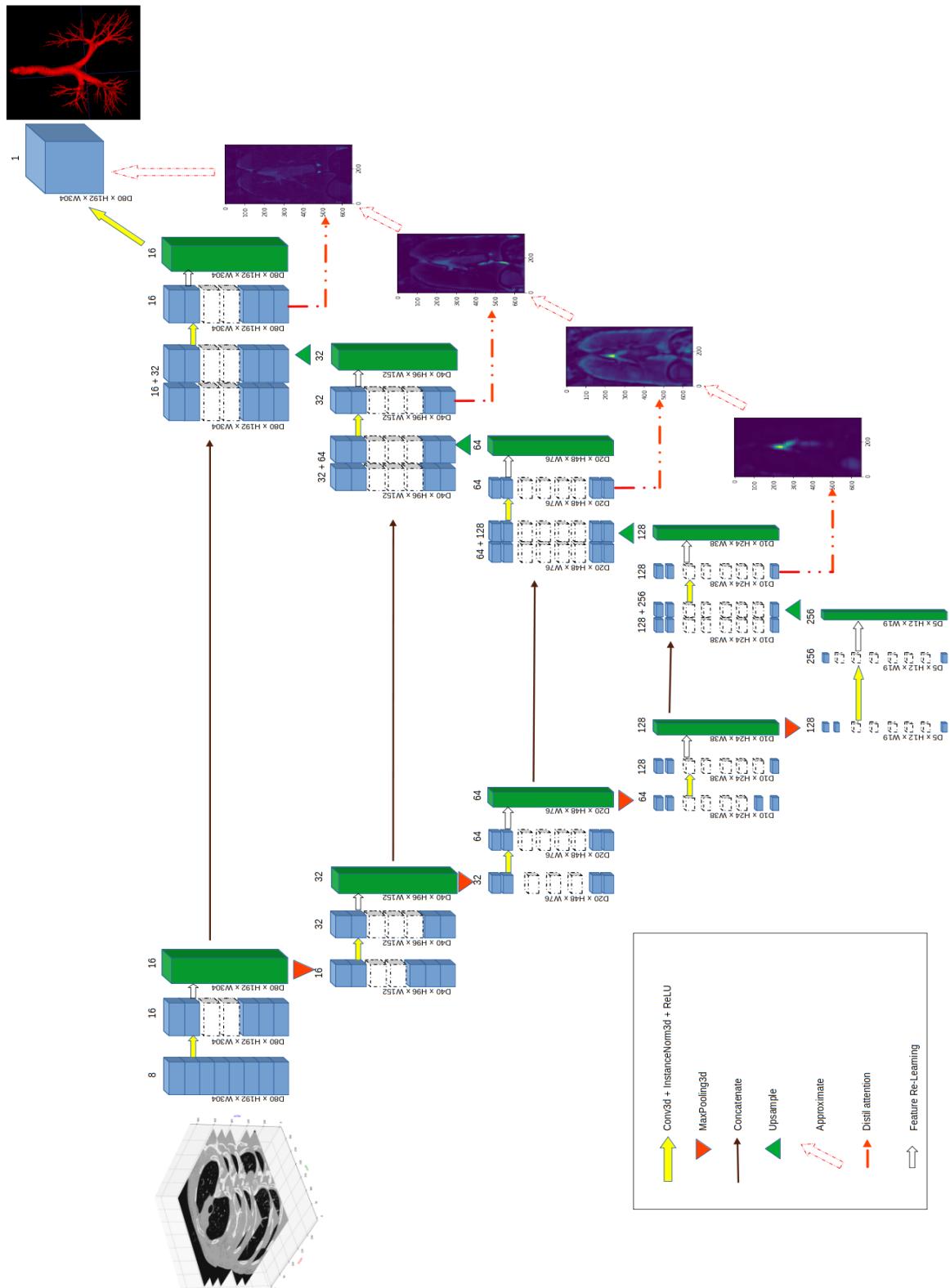


图 4-4 3D-UNet + Attention Distillation + Feature Re-Learning 网络结构示意图

Figure 4-4 The architecture of 3D-UNet + Attention Distillation + Feature Re-Learning network

的特征相同的 Shape。因此，3D-UNet 网络继续按照原来的方式工作。通道级的特征再学习模块将卷积层特征的通道先压缩再恢复，将多个通道进行重组。这样，有信息性的通道得到加强，而无信息性的通道就被抑制削弱，分割的性能得到提升。

图4-4的网络结构也是我们最终的网络结构模型，在3D-UNet 网络基准模型的基础上加上我们所做的两个改进模块：通道级的特征再学习模块和注意力蒸馏模块。我们称这个新的网络结构为 3D-UNet + Attention Distillation + Feature Re-Learning，简写为 3D-UNet + AD + FRL 网络结构。

### 4.3 综合实验

有了前述两章对 3D-UNet 网络基准模型的改进，我们得到了最终的改进的网络模型 3D-UNet + AD + FRL。现在我们来做一次对比实验，也是最终的综合性实验，看看支气管气道树最终的分割效果如何，综合评估分割模型的性能表现。

我们还是采用跟2.5.1节的 3D-UNet 网络基准模型的实验同样的训练集、验证集、测试集的数据，相同的实验条件。我们在上海交通大学高性能计算中心的 AI 超算上使用了 8 块 NVIDIA Tesla V100 显卡来同时进行训练，开启了 8 个进程，设置 Batch-Size=8，训练了 60 个迭代周期，验证和测试各 1 个迭代周期，整个实验耗时 39 小时 22 分钟。

集中验证集和测试集，计算它们的评价指标数据，如表4-1所示。在每一项指标上用双下划线加粗黑体标示出表现最优秀的，用波浪线标示出最差的。并在最后一行计算出每个指标的平均值，以作为参考。

我们从表4-1挑选出 9 个有代表性的病例来显示支气管气道树三维模型，见图4-5。

可以看到，末梢支气管的分割比之前的两个实验的效果要好得多，虽然不能完全排除红色假阴性（漏检）体素和绿色假阳性（误检）体素，总体上将末梢支气管这些精细特征都给分割出来了，说明通道级特征再学习的改进方法提高了 3D-UNet + AD + FRL 网络模型的性能。当然这些改善是注意力蒸馏 Attention Distillation 和通道级特征再学习 Channel-wise Feature Re-Learning 两个改进功能共同作用的，单一的改进还达不到这样的效果。

表 4-1 综合实验验证集和测试集的评价指标一览  
 每一项指标栏用双下划线加粗黑体标示出表现最优秀的, 用波浪线标示出表现最差的.

Table 4-1 The metrics overview of comprehensive experiment for valset and testset

	FPR	FNR	Sensitivity	Precision	DSC	BD	TLD
ATM_001_0000	<b>0.006</b>	7.119	92.88	<b>98.18</b>	95.04	<u><u>66.84</u></u>	82.96
ATM_024_0000	0.031	<u><u>7.676</u></u>	<u><u>92.32</u></u>	91.01	91.66	84.18	92.28
ATM_034_0000	0.023	3.712	96.29	96.00	96.06	88.44	93.92
ATM_041_0000	0.046	4.59	95.41	87.19	90.65	79.27	88.92
ATM_054_0000	0.031	4.924	95.08	92.47	93.76	70.28	<u><u>82.27</u></u>
ATM_055_0000	0.035	4.141	95.86	91.31	93.24	80.72	88.63
ATM_057_0000	0.034	5.475	94.53	91.42	92.94	74.57	85.34
ATM_060_0000	0.023	3.4	96.6	93.67	94.96	83.76	90.9
ATM_061_0000	0.027	4.291	95.71	92.16	93.62	79.23	88.00
ATM_074_0000	0.055	3.997	96.00	86.13	90.8	82.23	89.15
ATM_075_0000	0.024	5.497	94.5	93.15	93.76	77.36	86.64
ATM_080_0000	0.03	4.487	95.51	92.27	93.86	76.57	88.32
ATM_091_0000	0.042	4.036	95.96	89.15	92.43	75.53	87.34
ATM_150_0000	<u><u>0.084</u></u>	3.823	96.18	<u><u>79.61</u></u>	<u><u>87.11</u></u>	70.35	87.09
ATM_158_0000	0.043	3.584	96.42	87.12	91.53	78.66	89.93
ATM_163_0000	0.034	4.417	95.58	92.17	93.85	78.39	90.1
ATM_245_0000	0.036	0.401	<b>99.6</b>	84.2	91.26	<b>100</b>	99.89
ATM_246_0000	0.039	<b>0.288</b>	99.39	82.94	90.42	<b>100</b>	<b>99.9</b>
ATM_260_0000	0.017	0.984	99.02	94.59	<b>96.57</b>	97.24	96.57
ATM_266_0000	0.018	2.352	97.65	93.35	95.33	96.89	96
ATM_271_0000	0.046	1.354	98.65	87.13	92.45	96	95.64
ATM_638_0000	0.014	3.294	96.71	95.3	96.26	91.16	95.2
ATM_688_0000	0.016	2.107	97.89	94.49	95.36	91.77	95.32
平均值	0.0328	3.737	96.249	90.653	93.171	83.454	90.883

#### 4.4 性能指标对比

完成了综合实验之后, 我们就性能指标与一些经典的网络结构进行对比。我们选择了 V-Net<sup>[57]</sup>, VoxResNet<sup>[58]</sup>和 Attention-Gated U-Net<sup>[59]</sup>来比较 FPR、DSC、BD 和 TLD 四个指标。我们也跟 State-of-the-art(SOTA) 的几个方法, 诸如 Wang Chenglong 等人<sup>[60]</sup>提出的使用空间全连接网络解决管状结构的分割问题, Antonio Juarez 等人提出一个 3D-UNet + GNN 联合型网络<sup>[61]</sup>用于支气管气道树的分割, Qin Yulei 等人<sup>[39]</sup>提出一个体素连接的 AirwayNet 方法解决精确的气道树分割, Antonio Juarez 等人<sup>[62]</sup>较

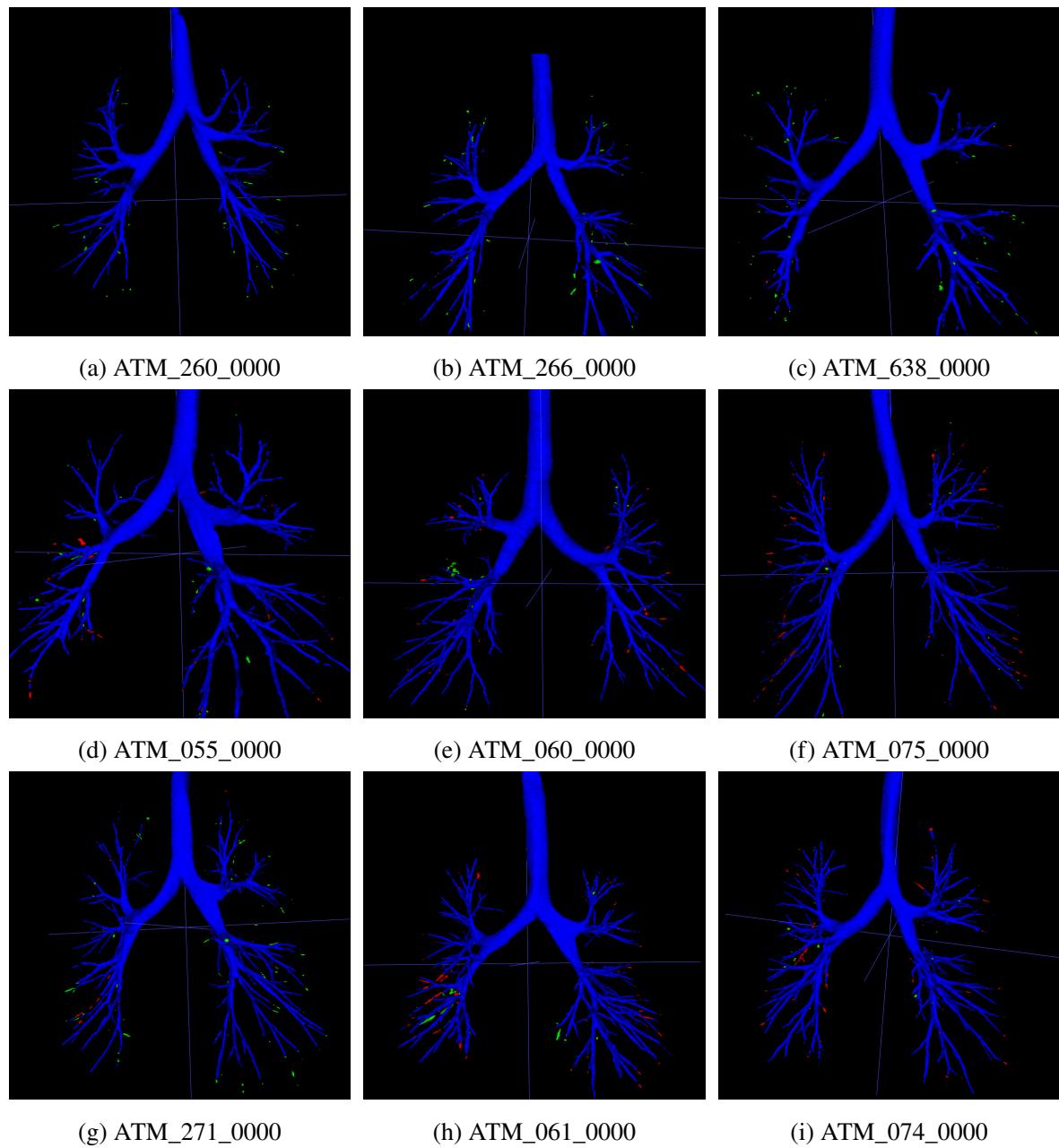


图 4-5 综合实验验证集和测试集的气道树分割效果一览

Figure 4-5 The airway tree segmentation overview of comprehensive experiment for valset and testset

早提出的应用 3D-UNet 网络用于支气管气道树分割，还有 Dakai Jin 等人<sup>[63]</sup>则是结合 3D CNN 与 Graph Refinement 在一个不完整标注的数据集上做的气道树分割，进行比较。我们选取 FPR、DSC、BD 和 TLD 指标的平均值来跟它们比较，见表4-2。可以看出，在 DSC 和 TLD 两项指标上，我们的方法要比经典的网络的方法要优秀一些，但

表 4-2 性能指标比较表

Table 4-2 The comparison table of performance metrics

方法		FPR (%)	DSC (%)	BD (%)	TLD (%)
我们的方法 3D-UNet + AD + FRL		0.0328	93.171	83.454	<u><b>90.883</b></u>
经典的 网络	V-Net <sup>[57]</sup>	0.024	92.1	91.0	81.6
	VoxResNet <sup>[58]</sup>	0.012	92.7	88.2	76.4
	Attention-Gated U-Net <sup>[59]</sup>	0.031	92.5	<u><b>93.8</b></u>	88.2
SOTA 的 方 法	Wang Chenglong 等人 <sup>[60]</sup> 提出的空间全连接网络	0.018	93.5	93.4	85.6
	Antonio Juarez 等人 <sup>[61]</sup> 提出的 3D-UNet + GNN 联合型网络	<u><b>0.009</b></u>	87.5	77.5	66.0
	Qin Yulei 等人 <sup>[39]</sup> 提出的体素连接 的 AirwayNet 网络	0.014	<u><b>93.7</b></u>	91.6	82.1
	Antonio Juarez 等人 <sup>[62]</sup> 较早提出的应用 3D-UNet 用于气道树分割的方法	0.014	93.6	91.9	80.7
	Dakai Jin 等人 <sup>[63]</sup> 提出的 结合 3D CNN 与 Graph Refinement 的方法	0.017	93.6	93.1	84.8
我们的方法在一些病例 上取得的优秀表现 以 ATM_260_0000 为例		0.017	96.57	97.24	96.57

跟 SOTA 的方法比，FPR、DSC 和 BD 三项指标都比较落后，只有 TLD 一项指标还保持着领先。当然，我们也在一些病例上取得了很优秀的结果。

## 4.5 本章小结

本章针对末梢支气管这样的精细特征引入了特征再学习功能。我们认为在不同的空间位置，特征应该具有不同的重要性。在枝干支气管这样的粗糙特征处赋予一般的权重，在末梢支气管这样的精细特征处应该赋予比较高的权重，而对于没有特征的背景或角落处则应该逐渐降低其权重。正是基于这样的一个想法，我们就想在卷积层块之后，最大池化层之前引入一种新的方法来调整特征的权重，从而想捕获

更精细的特征。这种新方法就是通道级特征再学习 (Channel-wise Feature Re-Learning, FRL) 方法。

我们详细介绍了通道级特征再学习方法的基本原理，提出了汇聚特征的计算公式。其中最重要的是特征立方体 (见图4-3) 的体积分过程，将体积分拆分为沿着 *Depth* 方向、*Height* 方向、*Width* 方向分别进行积分。这种积分是将特征立方体  $\text{Feat}_m$  从  $D \times H \times W$  大小挤压成  $1 \times 1 \times W$ ,  $1 \times H \times 1$  和  $D \times 1 \times 1$  的形状。这种挤压过程实际上是两次累积求和过程，它是借鉴了注意力蒸馏 (见图3-2) 的思想，就是想在空间的三个维度上都各自产生聚焦。将这三个方向的积分加起来 (见式4-9) 后，为了将多个通道重新组合，加强有信息的通道，抑制无信息的通道，我们又设计了一个降通道的卷积层和恢复原通道数的卷积层。两个卷积层用来处理三个方向的积分之和的数据得到  $\Psi(\text{Feat}_m)$  (见4-11式)，其跟卷积层输出的特征  $\text{Feat}_m$  在维数上是一样的，都是  $B \times C \times D \times H \times W$  的大小，这样我们就可以进行逐元素相乘。从图4-2、图4-3，公式4-3到公式 4-11是特征再学习的完整计算过程。特征再学习方法起的作用是在训练过程中、重要的气道特征随着特征再学习模块所赋予的新权重而逐渐被重视，而无关的区域则随着权重的降低而慢慢被忽略。

了解了通道级特征再学习的原理、计算过程和作用后，我们对 3D-UNet 网络基准基准模型进行改进，引入特征再学习模块，将其插入在每一层的卷积层块之后和最大池化层之间，见图4-4中的绿色柱体。使用 Python 实现了这个改进的 3D-UNet + Attention Distillation + Feature Re-Learning 网络之后，我们进行了一次综合实验。实验结果表明注意力蒸馏模块和特征再学习模块明显帮助提高了模型的性能，从表4-5看到分割出来的三维气道树模型表现为很完整的一棵树，枝干支气管和末梢支气管都已经完整分割出来。

完成综合实验，我们还做了指标对比研究。将我们的方法与经典的网络 V-Net, VoxResNet 和 Attention-Gated U-Net 比较 FPR、DSC、BD、TLD 四个指标，在 DSC 与 TLD 两项指标上取得领先。我们也跟 State-of-the-art (SOTA) 的方法进行对比，虽然在 TLD 指标上取得明显领先，但在 FPR、BD 两项指标确实明显落后。当然，我们的方法也在一些病例上取得了优秀的表现。

至此，本论文完成所有的研究与实验。

## 第五章 总结与展望

### 5.1 全文工作总结

本文研究肺部支气管气道树的分割技术，对于 CT 扫描图像数据，采用 3D-UNet 卷积网络进行监督学习的图像分割。本文首先介绍研究的背景和意义，肺部支气管气道树分割的研究现状和发展趋势，指出支气管气道树分割问题目前面临的两大挑战：1) 支气管体素在 CT 图像中的比例太低，造成了严重的类别不平衡；2) 全局尺寸和局部尺寸的特征信息差异悬殊，因支气管的树状结构决定了枝干支气管的粗糙特征和末梢支气管的精细特征同时存在，卷积层和池化层难以同时学习到这两种差异巨大的特征。针对这些挑战，本文提出 2 种方法，即注意力蒸馏方法和通道级特征再学习方法，来解决这些问题，它们也是本文的两大创新点。

本文围绕以下三点展开研究：

#### 1. 以 3D-UNet 网络作为基准分割模型

我们为 3D-UNet 基准网络设计了 9 个卷积层块，下采样路径每个卷积层块含有 2 个 Conv3d 卷积层、2 个 InstanceNorm3d 归一化层和 2 个 ReLU 激活函数，相邻卷积层块用最大池化层 MaxPool3d 相连（见图2-6）。上采样路径每个反卷积层块含有 2 个 Conv3d 反卷积层，相邻反卷积层块则用 Upsample 层相连（见图2-7）。3D-UNet 网络（结构如图2-8）从 ATM22 数据集读入切割成  $D80 \times H192 \times W304$  的小立方体数据，使用 Dice Loss 损失函数和 Focal Loss 损失函数计算预测值与真实值之差。我们在上海交通大学高性能计算中心的 AI 超算上进行训练，训练了 60 个迭代周期。我们从假阳性率 FPR、假阴性率 FNR、灵敏度 Sensitivity、精度 Precision、DSC、检测到的分支 BD 和 j 检测到的树长 TLD 共 7 个指标来评价分割的性能和效果。

#### 2. 提出注意力蒸馏方法改进基准模型

3D-UNet 基准网络对支气管气道树的分割在假阴性率上表现不佳，我们受注意力转移和知识蒸馏的启发，提出一种注意力蒸馏的方法来提高对支气管气道树特征的聚焦。注意力蒸馏是沿着通道维累积 C 个特征立方体，蒸馏浓缩到 1 个特征立方体（见图3-2），起到聚焦于支气管主目标的作用。我们设计实现了注意力蒸馏模块，添加到上采样路径的反卷积层块之后，以增加一个额外的梯度。如何计算这额外增加的梯度的损失，我们给出了注意力蒸馏的梯度损失算法3-1。

可视化注意力蒸馏后的图片效果（见图3–3和图3–3），可以观察到注意力的焦点在逐渐转移，从粗大的气管转移到细小的支气管。经过注意力蒸馏方法改进后的3D-UNet网络，进行了一次对比实验，实验结果表明抑制住了一些假阴性率，提高了分割模型的性能。

### 3. 采用通道级特征再学习方法提取精细特征

3D-UNet基准网络经过注意力蒸馏方法改进后，还存在一些不足，模型对末梢支气管的分割能力还不够。我们提出了第2种改进方法：通道级特征再学习方法。我们详细阐述例特征再学习方法的基本原理，计算方法与过程。特征再学习方法的重点在于对特征立方体  $D \times H \times W$  分别沿着 Depth、Height 和 Width 方向进行积分，挤压成  $D \times 1 \times 1$ ,  $1 \times H \times 1$  和  $1 \times 1 \times W$  的形状（见图4–3）。这种挤压实际上是一种两次累积求和的过程，它借鉴了注意力蒸馏的思想。我们还为特征再学习方法设计了2个卷积操作，为了是将多个通道进行重组，让有信息性的通道得到加强，而无信息性的通道就被抑制削弱。通道级特征再学习方法起的作用是在训练过程中，使重要的气道（粗糙的枝干支气管和精细的末梢支气管）增加权重逐渐被重视，而无关区域则降低权重而慢慢被忽略。

我们对3D-UNet + AD网络再次改进，加入通道级特征再学习功能模块（见图4–4），并进行一次综合实验。综合实验结果（见表4–5）表明气道树三维模型在假阴性率和假阳性率都得到了抑制，分割得到了完整清晰的气道树。完成了综合实验后，我们跟经典的网络方法对比了FPR、DSC、BD和TLD性能指标。我们的方法在DSC和TLD两项指标上领先，但跟SOTA的方法相比则在FPR、DSC和BD指标上落后了。

## 5.2 研究展望

本文针对肺部支气管气道树的分割以3D-UNet网络为基准模型，提出了2项改进方法：注意力蒸馏方法和通道级特征再学习方法。这些改进提高了分割模型的性能，但相比于SOTA的模型，我们落后了。因此本研究的第一个展望就是：

### A. 引入新的机制，继续改进，力争超过SOTA的方法。

支气管是一种树状结构，以每一个分支作为分割对象，可以设计成一种树网络模型。每扩展出一个分支，相应地树的深度增加1，分支与分支之间的连接就构成了树的一条边。以这样的树网络模型对支气管气道树进行分割。另外受NLP的注意力机制对Transformer巨大成功的鼓舞，我们在后面将引入Transformer来分割医疗图像，其

将会大幅提高支气管气道树的分割性能与效果。我们要继续改进，力争要超过 SOTA 的方法。

#### B. 应用本研究成果帮助开发导管手术机器人

本文在1.2小节就提到本研究的应用前景，我们要开发支气管导管手术机器人。本研究的一个直接动因就是为了开发导管手术机器人，分割出来的精密的气道树三维模型可提供给导管手术机器人，引导手术器械在患者肺部支气管里导航至目标的病灶部位。我们的研究使命是治疗疾病，帮助患者恢复健康。

#### C. 扩展到更多的管状对象的分割

与支气管这样的管状器官或组织类似，人体内还有很多，诸如肺部动脉/静脉血管，心脏的冠脉血管等。我们拟将本文中的 3D-UNet + AD + FRL 网络模型扩展到肺部血管的分割。后面还会将其扩展到心脏血管的分割中去，这会给心血管疾病的诊断和治疗带来很大的帮助，其中冠脉支架的植入就需要精确的血管分割，帮助定位血管栓塞和硬化的位置。

#### D. 引入弱监督学习和迁移学习，解决沉重的标注负担

医疗图像的标注是一项耗时、费力枯燥且成本高昂的工作，目前大量的医疗图像非常缺乏高质量的标注。如何解决沉重的标注负担，我们将研究引入弱监督学习和迁移学习，配合少量的标注来完成较大批量的自动分割任务。这将大大降低医疗负担，帮助临床医生更好地诊断和治疗。



## 参考文献

- [1] Fetita C I, Prêteux F, Beigelman-Aubry C, et al. Pulmonary airways: 3-D reconstruction from multislice CT and clinical investigation[J]. IEEE transactions on medical imaging, 2004, 23(11): 1353-1364.
- [2] Howling S, Evans T, Hansell D. The significance of bronchial dilatation on CT in patients with adult respiratory distress syndrome[J]. Clinical radiology, 1998, 53(2): 105-109.
- [3] Shaw R, Djukanovic R, Tashkin D, et al. The role of small airways in lung disease[J]. Respiratory medicine, 2002, 96(2): 67-80.
- [4] Wu X, Kim G H, Salisbury M L, et al. Computed tomographic biomarkers in idiopathic pulmonary fibrosis. The future of quantitative analysis[J]. American journal of respiratory and critical care medicine, 2019, 199(1): 12-21.
- [5] Li Y, Dai Y, Duan X, et al. Application of automated bronchial 3D-CT measurement in pulmonary contusion complicated with acute respiratory distress syndrome[J]. Journal of X-ray Science and Technology, 2019, 27(4): 641-654.
- [6] Estépar R S J, Kinney G L, Black-Shinn J L, et al. Computed tomographic measures of pulmonary vascular morphology in smokers and their clinical implications[J]. American journal of respiratory and critical care medicine, 2013, 188(2): 231-239.
- [7] Vigué J, Orte E M. 人体图谱·解剖学·组织学·病理学[M]. Ed. by 李云庆. 郑州: 河南科学技术出版社, 2012: 366-369.
- [8] Chand J. Slicer's Coordinate Systems[EB/OL]. [2021-12-29]. [https://www.slicer.org/wiki/Coordinate\\_systems](https://www.slicer.org/wiki/Coordinate_systems).
- [9] Adaloglu N. Understanding coordinate systems and DICOM for deep learning medical image analysis[J/OL]. <Https://theaisummer.com/>, 2020. <Https://theaisummer.com/medical-image-coordinates/>.
- [10] Company I S. ION bronchoscopy robot[EB/OL]. <https://www.intuitive.com/en-us/products-and-services/ion>.
- [11] Otsu N. A threshold selection method from gray-level histograms[J]. IEEE transactions on systems, man, and cybernetics, 1979, 9(1): 62-66.

- [12] MacQueen J. Some methods for classification and analysis of multivariate observations[C]// Proc. 5th Berkeley Symposium on Math., Stat., and Prob. 1965: 281.
- [13] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
- [14] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation[C]// Navab N, Hornegger J, Wells W M, et al. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham: Springer International Publishing, 2015: 234-241.
- [15] Milletari F, Navab N, Ahmadi S A. V-net: Fully convolutional neural networks for volumetric medical image segmentation[C]// 2016 fourth international conference on 3D vision (3DV). 2016: 565-571.
- [16] Zhou Z, Siddiquee M M R, Tajbakhsh N, et al. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation[J]. IEEE transactions on medical imaging, 2019, 39(6): 1856-1867.
- [17] Çiçek Ö, Abdulkadir A, Lienkamp S S, et al. 3D U-Net: learning dense volumetric segmentation from sparse annotation[C]// Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19. 2016: 424-432.
- [18] Sauvola J, Pietikäinen M. Adaptive document image binarization[J]. Pattern recognition, 2000, 33(2): 225-236.
- [19] Mustra M, Delac K, Grgic M. Overview of the DICOM standard[C]// 2008 50th International Symposium ELMAR: vol. 1. 2008: 39-44.
- [20] Noh H, Hong S, Han B. Learning Deconvolution Network for Semantic Segmentation[J]. 2015 IEEE International Conference on Computer Vision (ICCV), 2015: 1520-1528.
- [21] Li X, Chen H, Qi X, et al. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes[J]. IEEE Transactions on Medical Imaging, 2017, 37: 2663-2674.

- [22] Dolz J, Desrosiers C, Ben Ayed I. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study[J]. NeuroImage, 2017, 170. DOI: 10.1016/j.neuroimage.2017.04.039.
- [23] Perslev M, Dam E B, Pai A, et al. One Network to Segment Them All: A General, Lightweight System for Accurate 3D Medical Image Segmentation[C] // Shen D, Liu T, Peters T M, et al. Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Cham: Springer International Publishing, 2019: 30-38.
- [24] Ullah Z, Usman M, Latif S, et al. Densely attention mechanism based network for COVID-19 detection in chest X-rays[J]. Scientific Reports, 2023, 13.
- [25] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. ArXiv, 2019, abs/1810.04805.
- [26] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[C] // Guyon I, Luxburg U V, Bengio S, et al. Advances in Neural Information Processing Systems: vol. 30. Curran Associates, Inc., 2017.
- [27] Hatamizadeh A, Tang Y, Nath V, et al. UNETR: Transformers for 3D Medical Image Segmentation[C] // 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2022: 1748-1758. DOI: 10.1109/WACV51458.2022.00181.
- [28] Fenimore C, Armato S, Aberle D, et al. The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans[J/OL]., 2011(38). [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=907229](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=907229).
- [29] Lo P, van Ginneken B, Reinhardt J M, et al. Extraction of Airways From CT (EX-ACT'09)[J]. IEEE transactions on medical imaging, 2012, 31 11: 2093-107.
- [30] Zhao T, Yin Z, Wang J, et al. Bronchus Segmentation and Classification by Neural Networks and Linear Programming[C] // International Conference on Medical Image Computing and Computer-Assisted Intervention. 2019.
- [31] Yun J, Park J, Yu D, et al. Improvement of fully automated airway segmentation on volumetric computed tomographic images using a 2.5 dimensional convolutional neural net[J]. Medical Image Analysis, 2019, 51: 13-20.

- [32] Meng Q, Roth H R, Kitasaka T, et al. Tracking and Segmentation of the Airways in Chest CT Using a Fully Convolutional Network[C]//International Conference on Medical Image Computing and Computer-Assisted Intervention. 2017.
- [33] Cheng G, Wu X, Xiang W, et al. Segmentation of the Airway Tree From Chest CT Using Tiny Atrous Convolutional Network[J]. IEEE Access, 2021, 9: 33583-33594. DOI: 10.1109/ACCESS.2021.3059680.
- [34] Guo J, Fu R, Pan L, et al. Coarse-to-fine Airway Segmentation Using Multi information Fusion Network and CNN-based Region Growing[J]. Computer methods and programs in biomedicine, 2021, 215: 106610.
- [35] Yu W, Zheng H, Gu Y, et al. TNN: Tree Neural Network for Airway Anatomical Labeling[J]. IEEE Transactions on Medical Imaging, 2023, 42(1): 103-118. DOI: 10.1109/TMI.2022.3204538.
- [36] Aurelio Y, De Almeida G, Castro C, et al. Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function[J]. Neural Processing Letters, 2019, 50. DOI: 10.1007/s11063-018-09977-1.
- [37] Qin Y, Zheng H, Gu Y, et al. Learning Tubule-Sensitive CNNs for Pulmonary Airway and Artery-Vein Segmentation in CT[J]. IEEE Transactions on Medical Imaging, 2021, 40(6): 1603-1617. DOI: 10.1109/TMI.2021.3062280.
- [38] Zhang M, Zhang H, Yang G Z, et al. CFDA: Collaborative Feature Disentanglement and Augmentation for Pulmonary Airway Tree Modeling of COVID-19 CTs[C]// Wang L, Dou Q, Fletcher P T, et al. Medical Image Computing and Computer Assisted Intervention – MICCAI 2022. Cham: Springer Nature Switzerland, 2022: 506-516.
- [39] Qin Y, Chen M, Zheng H, et al. AirwayNet: A Voxel-Connectivity Aware Approach for Accurate Airway Segmentation Using Convolutional Neural Networks[C]// Shen D, Liu T, Peters T M, et al. Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Cham: Springer International Publishing, 2019: 212-220.
- [40] Yu W, Zheng H, Zhang M, et al. Break: Bronchi Reconstruction by Geodesic Transformation and Skeleton Embedding[C]//2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). 2022: 1-5. DOI: 10.1109/ISBI52829.2022.9761697.

- [41] Zheng H, Qin Y, Gu Y, et al. Alleviating Class-Wise Gradient Imbalance for Pulmonary Airway Segmentation[J]. IEEE Transactions on Medical Imaging, 2021, 40(9): 2452-2462. DOI: 10.1109/TMI.2021.3078828.
- [42] Kamaruddin N, Rajion Z A, Yusof A, et al. Relationship between Hounsfield unit in CT scan and gray scale in CBCT[J/OL]. AIP Conference Proceedings, 2016, 1791(1): 020005. eprint: <https://aip.scitation.org/doi/pdf/10.1063/1.4968860>. <https://aip.scitation.org/doi/abs/10.1063/1.4968860>. DOI: 10.1063/1.4968860.
- [43] Chollet F. Python 深度学习[M]. Ed. by 张亮. 北京: 人民邮电出版社, 2018: 99.
- [44] Ba J, Kiros J R, Hinton G E. Layer Normalization[J]. ArXiv, 2016, abs/1607.06450.
- [45] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. ArXiv, 2015, abs/1502.03167.
- [46] Ulyanov D, Vedaldi A, Lempitsky V S. Instance Normalization: The Missing Ingredient for Fast Stylization[J]. ArXiv, 2016, abs/1607.08022.
- [47] Wu Y, He K. Group Normalization[J]. International Journal of Computer Vision, 2018, 128: 742-755.
- [48] Qiao S, Wang H, Liu C, et al. Weight Standardization[J]. ArXiv, 2019, abs/1903.10520.
- [49] Qiao S, Wang H, Liu C, et al. Rethinking Normalization and Elimination Singularity in Neural Networks[J]. ArXiv, 2019, abs/1911.09738.
- [50] Zare M, Rohatgi S. DeepNorm-A Deep Learning Approach to Text Normalization[J]. ArXiv, 2017, abs/1712.06994.
- [51] Lin T Y, Goyal P, Girshick R B, et al. Focal Loss for Dense Object Detection[J]. 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 2999-3007.
- [52] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. CoRR, 2014, abs/1412.6980.
- [53] team S H. 上海交大超算平台用户手册[J/OL], 2021. <https://docs.hpc.sjtu.edu.cn/system/index.html>.
- [54] Yushkevich P, Piven J, Hazlett H, et al. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability[J]. NeuroImage, 2006, 31(3): 1116-1128. DOI: 10.1016/j.neuroimage.2006.01.015.

- [55] Zagoruyko S, Komodakis N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer[J]. ArXiv, 2016, abs/1612.03928.
- [56] Hinton G E, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[J]. ArXiv, 2015, abs/1503.02531.
- [57] Milletarì F, Navab N, Ahmadi S A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation[J]. 2016 Fourth International Conference on 3D Vision (3DV), 2016: 565-571.
- [58] Chen H, Dou Q, Yu L, et al. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images[J/OL]. NeuroImage, 2018, 170: 446-455. <https://www.sciencedirect.com/science/article/pii/S1053811917303348>. DOI: <https://doi.org/10.1016/j.neuroimage.2017.04.041>.
- [59] Oktay O, Schlemper J, Folgoc L L, et al. Attention U-Net: Learning Where to Look for the Pancreas[J]. ArXiv, 2018, abs/1804.03999.
- [60] Wang C, Hayashi Y, Oda M, et al. Tubular Structure Segmentation Using Spatial Fully Connected Network with Radial Distance Loss for 3D Medical Images[C]// Shen D, Liu T, Peters T M, et al. Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Cham: Springer International Publishing, 2019: 348-356.
- [61] Garcia-Uceda Juarez A, Selvan R, Saghir Z, et al. A Joint 3D UNet-Graph Neural Network-Based Method for Airway Segmentation from Chest CTs[C]// Suk H I, Liu M, Yan P, et al. Machine Learning in Medical Imaging. Cham: Springer International Publishing, 2019: 583-591.
- [62] Garcia-Uceda Juarez A, Tiddens H A W M, de Bruijne M. Automatic Airway Segmentation in Chest CT Using Convolutional Neural Networks[C]// Stoyanov D, Taylor Z, Kainz B, et al. Image Analysis for Moving Organ, Breast, and Thoracic Images. Cham: Springer International Publishing, 2018: 238-250.
- [63] Jin D, Xu Z, Harrison A P, et al. 3D Convolutional Neural Networks with Graph Refinement for Airway Segmentation Using Incomplete Data Labels[C]// Wang Q, Shi Y, Suk H I, et al. Machine Learning in Medical Imaging. Cham: Springer International Publishing, 2017: 141-149.

## 附录 A 实验的部分源代码

以下为实验中编写的部分源代码，以供复现实验结果或者帮助理解论文中的一些关键的计算过程与算法。

为复现注意力蒸馏后的效果，请查看以下一些代码片段：

**Listing A.1 Attention Distillation Effect**

```

1
2 import numpy as np
3 import SimpleITK as sitk
4 from matplotlib import pyplot as plt
5
6 def load_CT_scan_3D_image(niigz_file_name):
7     itkimage = sitk.ReadImage(niigz_file_name)
8     numpyImages = sitk.GetArrayFromImage(itkimage)
9     numpyOrigin = np.array(list(reversed(itkimage.GetOrigin())))
10    numpySpacing = np.array(list(reversed(itkimage.GetSpacing())))
11    return numpyImages, numpyOrigin, numpySpacing
12
13 #=====
14
15 ATM_074_0000_niigz_files = ["ATM_074_0000-feat6.nii.gz",
16                               "ATM_074_0000-feat7.nii.gz",
17                               "ATM_074_0000-feat8.nii.gz",
18                               "ATM_074_0000-feat9.nii.gz"]
19
20 feat6_image_npy, feat6_origin, feat6_spacing =
21     load_CT_scan_3D_image(ATM_074_0000_niigz_files[0])
22
23 feat7_image_npy, feat7_origin, feat7_spacing =
24     load_CT_scan_3D_image(ATM_074_0000_niigz_files[1])
25
26 feat8_image_npy, feat8_origin, feat8_spacing =
27     load_CT_scan_3D_image(ATM_074_0000_niigz_files[2])
28
29 feat9_image_npy, feat9_origin, feat9_spacing =
30     load_CT_scan_3D_image(ATM_074_0000_niigz_files[3])
31
32
33 depth, height, width = feat6_image_npy.shape
34
35 plt.figure(figsize=(10, 10))
36 plt.imshow(np.flipud(feat6_image_npy[:, height//2, :]))
37 plt.imshow(np.flipud(feat6_image_npy[:, height//2, :]), cmap="gray")
38
39 plt.figure(figsize=(10, 10))

```

```

40 plt.imshow(np.flipud(feat7_image_npy[:, height//2, :]))
41 plt.imshow(np.flipud(feat7_image_npy[:, height//2, :]), cmap="gray")
42
43 plt.figure(figsize=(10, 10))
44 plt.imshow(np.flipud(feat8_image_npy[:, height//2, :]))
45 plt.imshow(np.flipud(feat8_image_npy[:, height//2, :]), cmap="gray")
46
47 plt.figure(figsize=(10, 10))
48 plt.imshow(np.flipud(feat9_image_npy[:, height//2, :]))
49 plt.imshow(np.flipud(feat9_image_npy[:, height//2, :]), cmap="gray")

```

通道级特征再学习方法的实现：

### Listing A.2 Channel-wise Feature Re-Learning Module

```

1
2 import torch.nn as nn
3
4 class FeatureReLearningModule(nn.Module):
5     def __init__(self, num_channels, Depth, Height, Width, reduction_ratio=2):
6         super().__init__()
7         num_reduced_channels = num_channels // reduction_ratio
8         self.reduction_ratio = reduction_ratio
9         self.conv_module = nn.Sequential(
10             nn.Conv3d(in_channels=num_channels,
11                       out_channels=num_reduced_channels,
12                       kernel_size=1,
13                       stride=1),
14             nn.ReLU(inplace=True),
15             nn.Conv3d(in_channels=num_reduced_channels,
16                       out_channels=num_channels,
17                       kernel_size=1,
18                       stride=1),
19             nn.Sigmoid())
20         self.spatial_dimension = [Depth, Height, Width]
21         self.Depth_squeeze = nn.Conv3d(in_channels=Depth,
22                                         out_channels=1,
23                                         kernel_size=1,
24                                         stride=1)
25         self.Height_squeeze = nn.Conv3d(in_channels=Height,
26                                         out_channels=1,
27                                         kernel_size=1,
28                                         stride=1)
29         self.Width_squeeze = nn.Conv3d(in_channels=Width,
30                                         out_channels=1,
31                                         kernel_size=1,
32                                         stride=1)
33     def forward(self, input_tensor):
34         squared_tensor = torch.pow(input_tensor, exponent=2)
35         # Weight along channels and different axes

```

```

36     Depth, Height, Width = self.spatial_dimension[0], \
37                               self.spatial_dimension[1], \
38                               self.spatial_dimension[2]
39     Depth_axis = input_tensor.permute(0, 2, 1, 3, 4)           # B, D, C, H, W
40     Height_axis = input_tensor.permute(0, 3, 2, 1, 4)          # B, H, D, C, W
41     Z_spatial_integration_on_Depth = \
42         self.Height_squeeze(Height_axis).permute(0, 4, 2, 1, 3)
43     Z_spatial_integration_on_Depth = \
44         self.Width_squeeze(Z_spatial_integration_on_Depth).permute(0, 4, 2, 3, 1)
45     Z_spatial_integration_on_Height = \
46         self.Depth_squeeze(Depth_axis).permute(0, 4, 1, 3, 2)
47     Z_spatial_integration_on_Height = \
48         self.Width_squeeze(Z_spatial_integration_on_Height).permute(0, 4, 2, 3, 1)
49     Z_spatial_integration_on_Width = \
50         self.Depth_squeeze(Depth_axis).permute(0, 3, 1, 2, 4)
51     Z_spatial_integration_on_Width = \
52         self.Height_squeeze(Z_spatial_integration_on_Width).permute(0, 3, 2, 1, 4)
53     Z_spatial_integration = Z_spatial_integration_on_Depth + \
54                               Z_spatial_integration_on_Height + \
55                               Z_spatial_integration_on_Width
56     channel_descriptor = self.conv_module(Z_spatial_integration)
57     recalibrated_feature = torch.mul(input_tensor, channel_descriptor)
58     feature_mapping = torch.sum(squared_tensor, dim=1, keepdim=True)
59     return recalibrated_feature, feature_mapping

```

在 AI 超算上申请作业的脚本代码:

Listing A.3 Airway3DSegment\_Baseline.slurm

```

1  #!/bin/bash
2
3
4  #SBATCH --job-name=UNet3D_Airway3DSegment
5  #SBATCH --partition=dgx2
6  #SBATCH -n 8
7  #SBATCH --gres=gpu:2
8  #SBATCH --output=Airway3DSegment_Baseline_%j.out
9  #SBATCH --error=Airway3DSegment_Baseline_%j.err
10 #SBATCH --mail-type=end
11 #SBATCH --mail-user=frederique.hsu@outlook.com
12
13 module load miniconda3
14
15 source activate ~/.conda/envs/my-dissertation-env
16
17 which python3
18
19 python3 --version
20
21 # Training the baseline UNet3D model

```

```
22 python3 Airway3DSegmentation.py --model=baseline --batch-size=4 --num-workers=3 \
23 --save-dir=baseline --train-cube-size 80 192 304 --train-stride 64 96 152 \
24 --val-cube-size 80 192 304 --val-stride 64 72 72 --start-epoch=1 --epochs=60 \
25 --sgd=0 --randsel=0 --multi-gpu-parallel=1 --encoder-path-ad=0 --decoder-path-ad=0 \
26 --enable-training=1 --enable-validating=0 --enable-testing=0
27
28
29 # Validating
30 python3 Airway3DSegmentation.py --model=baseline --batch-size=4 --num-workers=3 \
31 --save-dir=baseline --train-cube-size 80 192 304 --train-stride 64 96 152 \
32 --val-cube-size 80 192 304 --val-stride 64 72 72 --start-epoch=1 --epochs=60 \
33 --sgd=0 --randsel=0 --multi-gpu-parallel=1 --encoder-path-ad=0 --decoder-path-ad=0 \
34 --resume=". ./results/baseline/model_latest.ckpt" --save-feature=1 \
35 --enable-training=0 --enable-validating=1 --enable-testing=0
36
37
38 # Testing
39 python3 Airway3DSegmentation.py --model=baseline --batch-size=4 --num-workers=3 \
40 --save-dir=baseline --train-cube-size 80 192 304 --train-stride 64 96 152 \
41 --val-cube-size 80 192 304 --val-stride 64 72 72 --start-epoch=1 --epochs=60 \
42 --sgd=0 --randsel=0 --multi-gpu-parallel=1 --encoder-path-ad=0 --decoder-path-ad=0 \
43 --resume=". ./results/baseline/model_latest.ckpt" --save-feature=1 \
44 --enable-training=0 --enable-validating=0 --enable-testing=1
```

本论文的全部代码开源在 GitHub Airway3DSegmentNet 仓库，若有需要请访问该源代码仓库。

## 致 谢

本论文经历了一个曲折的过程，历经千辛万苦走到今天终于完成了。

现在回过头来回顾过去，心中有很多感慨，有许多感激之言。首先，我要感谢卢宏涛教授、刘维平副教授两位导师对我的论文指导，也要感谢我的上任导师朱弘恣教授（我的毕业论文经历过一次换课题换导师）。

我要特别感谢上海交通大学高性能计算中心的费晓舒老师、郭武老师，感谢他们免费给我提供宝贵的计算资源（给我分配了 8 张 NVIDIA Tesla V100 显卡 512GB HBM2 显存，2 颗 Intel Xeon 铂金 8168 CPU, 1.5TB DDR4 共享内存, 30TB NVMe SSD 共享存储）和全程的技术支持。本论文是在上海交通大学高性能计算中心的  $\pi$ 2.0 AI 超算上完成计算任务的。若没有费晓舒、郭武两位老师的大力支持，我根本无法完成此论文。

我还要感谢秦玉磊学长、谢昭智博士的指导和帮助，感谢吴娜学姐的督促和支持帮助。我要感谢前同事丁小柳先生、李志坚先生的帮助和技术指导。

感谢张忠能老师、姚天昉老师的帮助和支持。

我要感谢我的父母和弟弟徐攀，非常感谢你们的支持和持续的监督。感谢余学琴姐姐的支持。

我要感谢自从 2016 年我考进上海交通大学硕士研究生以来帮助和支持过我学业的老师和同学。

另外，本论文的撰写使用了开源的 SJTUThesis 上海交通大学学位论文 L<sup>A</sup>T<sub>E</sub>X 排版模板，感谢 @weijianwen 学长一直以来的开发和维护工作。感谢 L<sup>A</sup>T<sub>E</sub>X 和 SJTUThesis，帮我自动排版，节省了大量时间。

最后衷心表示我的感谢！



## 学术论文和科研成果目录

### 学术论文

- [1] 徐赞. 肺部动静脉血管分割技术研究. 上海交通大学计算机科学与工程系微信公众号公示, 2023.03