深圳大学考试答题纸

(以论文、报告等形式考核专用)  
二○二 四 ～二○二 五 学年度第 1 学期

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | 15019900006 | | 课程名称 | 人工智能导论 | | 主讲教师 | 高灿 | 评分 |  |
| 学 号 | 2022150221 | | 姓名 | 何泽锋 | 专业年级 | 2022级计算机科学与技术（高性能班） | | | |
|  | | | | | | | | | |
| 教师评语： | | | | | | | | | |
| 题目： | | 基于U-Net的肝脏分割系统 | | | | | |  | |

**基于U-Net的肝脏分割系统**

**【摘要】**

在医学影像分析领域，肝脏分割技术的发展对于疾病的诊断和治疗规划至关重要。随着深度学习技术的进步，尤其是U-Net架构[1]的出现，医学图像分割领域迎来了革命性的变化。U-Net因其独特的结构和优势，在医学图像分割任务中表现出色。U-Net网络的设计允许输入任何尺寸的数据，并且能够实现像素级别的图像分割，这使得其在提高分割精确率方面具有显著优势。U-Net的结构简单适用于医学数据较少的情况此外，U-Net通过跳跃连接有效地融合来自浅层和深层的特征，使得模型既能捕捉图像的全局语义信息，又不丢失细粒度的局部信息。这种结构上的优势使得U-Net在医学图像分割领域成为最常用的深度学习架构之一。

因此，本文将介绍一个基于U-Net的肝脏分割模型，该模型利用U-Net架构的优势，通过深度学习技术处理医学影像中的肝脏分割问题，旨在展示基于U-Net的肝脏分割系统在医学影像分析中的效果。

【**正文**】

1. **简介及相关工作**

**1.1 简介**

肝脏分割作为医学图像处理领域的核心任务之一，对于提高肝脏疾病诊断的准确性和治疗规划的有效性至关重要。近年来，深度学习技术的迅猛发展，尤其是卷积神经网络（CNN）的广泛应用，为肝脏分割带来了革命性的进步。CNN的强大特征提取能力使其在处理复杂的医学图像数据时表现出色。本研究旨在开发一个基于深度学习的肝脏分割系统，该系统利用U-Net架构，这是一种专为图像分割任务设计的网络结构，以其优异的性能在多个医学图像分割任务中取得了显著成果。

本系统的核心目标是实现对肝脏区域的高精度分割，以辅助医生进行更准确的疾病诊断和治疗决策。通过深入研究和优化U-Net模型，系统能够从图片中识别并分割出肝脏，同时提供直观的分割结果和定量的评估指标，以衡量分割质量。

**1.2 相关工作**

在医学图像分割领域，尤其是肝脏分割，已经有许多研究工作。早期的方法主要依赖于传统的图像处理技术，如阈值分割、区域生长和主动轮廓模型等。然而，这些方法在处理具有复杂背景和不规则形状的肝脏时存在局限性。随着深度学习技术的兴起，基于CNN的方法逐渐成为主流。U-Net架构因其在多个医学图像分割任务中的卓越表现而受到广泛关注。U-Net通过编码器-解码器结构，结合跳跃连接，有效捕捉图像的上下文信息和位置信息，从而实现精确的分割。

在接下来的章节中，将详细介绍本系统的原理和方法，包括U-Net架构的实现、数据预处理、模型训练和评估等关键步骤。同时，也将展示实验结果，以证明本系统在肝脏分割任务中的有效性。

**1.3 实验环境**

本实验基于PyTorch深度学习框架实现。PyTorch广泛用于计算机视觉和自然语言处理领域。选择PyTorch作为实验框架的原因在于其动态计算图的特性，这使得模型的构建、调试和实验变得更加灵活和直观。PyTorch的自动微分系统能够自动计算梯度，极大地简化了深度学习模型的训练过程。此外还提供了良好的GPU支持，对于需要大量计算资源的肝脏分割模型训练尤为重要，能够有效地加速训练过程。

在图像处理方面，实验中使用了OpenCV和Pillow库。OpenCV提供了广泛的图像处理和计算机视觉功能，包括图像的读取、显示、转换和特征提取等。Pillow提供了简单的图像处理能力，如图像的加载、保存和基本的图像转换操作。

数据处理和科学计算方面，实验中使用了NumPy和Matplotlib库。NumPy提供了高性能的多维数组对象和一系列操作这些数组的函数。Matplotlib可将结果可视化，更直观地理解模型训练过程中的变化和结果。

为了更好地进行深度学习模型的训练和评估，实验中还使用了Torchvision库。Torchvision是PyTorch的官方图像处理库，它提供了图像转换操作和预训练模型，与PyTorch深度集成，方便模型训练和评估。

1. **原理及方法分析**

**2.1 数据预处理**

在肝脏分割任务中，数据预处理是一个关键步骤，直接影响模型的训练效果和最终性能。在数据处理部分，定义了一个LiverDataset类，该类继承自torch.utils.data.Dataset，用于加载和处理训练及验证数据。具体步骤如下：

**2.1.1数据加载**

数据加载是预处理的第一步，从指定的目录中读取图像及其对应的标签。make\_dataset函数从指定的目录中读取图像及其对应的标签（ground truth），此处展示其中一组图像及其标签，左侧为原始CT图像，右侧为肝脏标签。

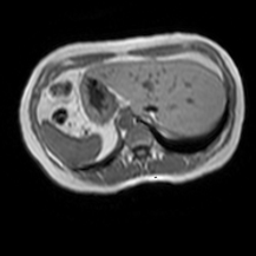
 

图1 原始CT图像 图2 标签图像

将读入的数据存储在一个列表中，图像和标签的路径分别从Data和Ground中获取，每个元素是一个包含图像和标签的元组。这种结构便于后续的模型训练。

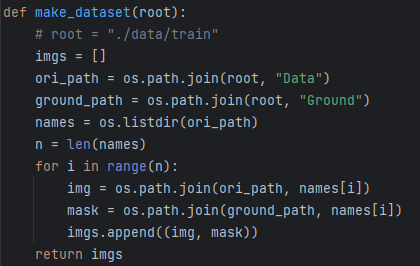


图3 数据读入

**2.1.2图像转换**

肝脏分割任务处理的是灰度图像，因此可以将图像转换为灰度图以减少计算量，并且对于分割任务来说颜色信息并不是必需的。因此，在\_\_getitem\_\_方法中，使用PIL库打开图像，将其转换为灰度图（'L'模式）。

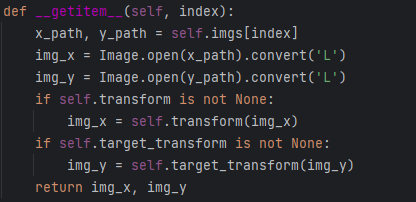


图4 转灰度图

**2.1.3 Tensor转换**

将PIL图像对象转换为PyTorch张量，通过transforms.ToTensor()实现转换，将[0, 255]范围的像素值归一化到[0.0, 1.0]。具体使用如下所示，此处的x\_transforms是transforms.ToTensor()，作为参数传入后，\_\_getitem\_\_会调用进行归一化。归一化像素值可以加快模型的训练速度，因为它确保了所有输入特征都在同一尺度上。

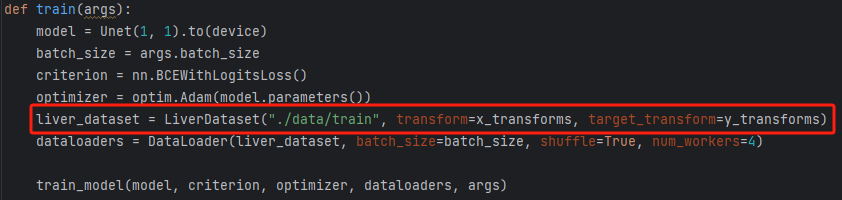


图5 tensor转换

**2.2 模型构建**

模型使用U-Net架构，采用编码器-解码器结构进行图像分割任务。它由一个收缩路径（左侧）和一个扩展路径（右侧）组成。编码器部分负责捕捉图像的上下文信息，通过一系列的卷积层和池化层逐步降低图像的空间维度，同时增加特征通道的数量，从而提取更深层次的特征。解码器部分负责恢复图像的空间维度，将编码器中提取的特征映射回原始图像的分辨率。这一过程通过上采样和特征融合来实现。

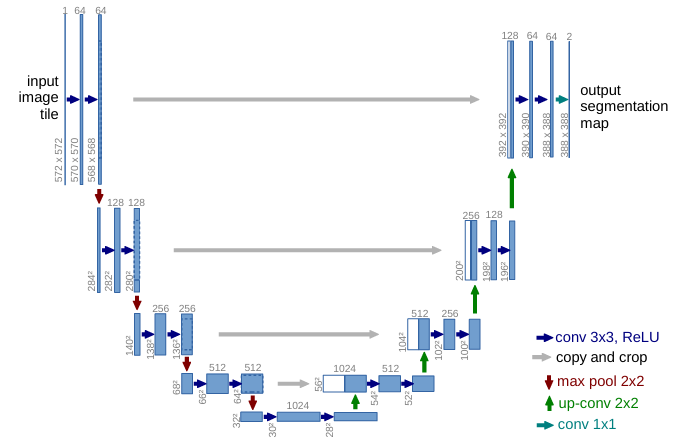


图6 U-Net模型图

**2.2.1下采样**

编码器包含多个DoubleConv块，每个DoubleConv块包含两个3x3的卷积层，这些卷积层之间是批量归一化（BatchNorm）和ReLU激活函数。在每个DoubleConv块之后，使用2x2的最大池化操作进行下采样，逐步降低特征图的空间维度，同时增加特征通道数。

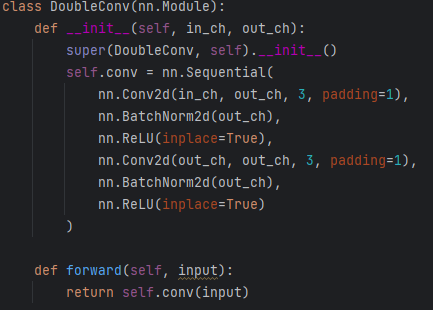
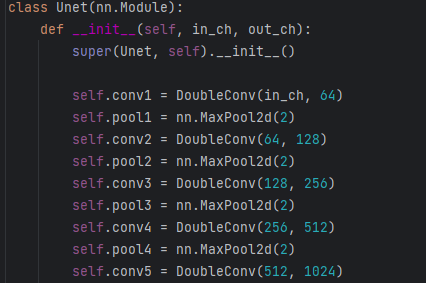
 

图7 卷积块 图8 下采样

**2.2.2 上采样**

解码器部分通过转置卷积ConvTranspose进行上采样，增加特征图的空间尺寸，同时减少特征通道的数量。在每次上采样后，将来自编码器的相应特征图与解码器的特征图进行拼接，恢复图像的细节信息。

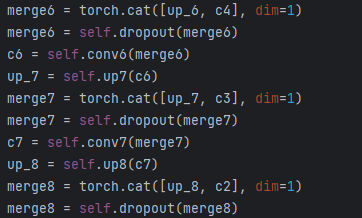


图9 上采样

**2.2.3 输出层**

通过一个1x1卷积层将特征图转换为与目标分割图相同数量的通道，这里为1个通道，代表二值分割图。



图10 输出层

**2.3 训练**

训练是一个迭代的过程，它涉及到模型权重的调整，以最小化损失函数并提高模型的性能。以下是训练U-Net模型时涉及的关键步骤：

**2.3.1 损失函数**

使用二元交叉熵损失BCEWithLogitsLoss计算模型输出和真实标签之间的差异。这种损失函数结合了Sigmoid激活函数和二元交叉熵损失，使得模型在输出端不需要额外的激活层，简化了模型结构。

**2.3.2 优化器**

使用Adam优化器来更新模型的权重，是一种自适应学习率的优化算法。通过计算梯度的一阶矩估计和二阶矩估计来调整每个参数的学习率，从而加速收敛并提高训练的稳定性。

**2.3.3 批处理和数据加载**

通过DataLoader类实现批处理，在每个epoch中迭代多个批次的数据，同时使用数据打乱和多核加载的功能。

**2.3.4模型训练**

模型训练阶段，遍历所有的训练数据批次，执行以下步骤：

·前向传播：模型对输入数据进行预测，计算输出。

·损失计算：根据模型输出和真实标签，计算损失函数的值。

·反向传播：根据损失值，计算模型权重的梯度。

·权重更新：使用优化器根据梯度更新模型的权重。

这个过程在每个epoch中重复进行，直到达到预定的epoch次数。

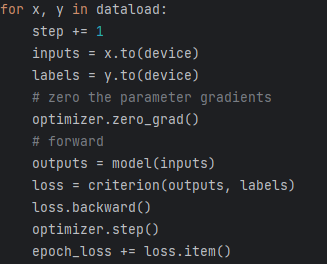


图11 模型训练

**2.3.5 模型保存**

早停法：在验证损失减少时保存模型的最佳状态，避免模型在训练集上过拟合，同时保留了在验证集上表现最佳的模型参数。



图12 保存最优模型

定期保存：定期保存模型的权重，以防训练过程中断导致的结果丢失。即使训练被意外中断，也能够从最近的保存点恢复。

****

图13 定期保存

1. **优化和性能提升**

**3.1 代码优化和性能提升**

在开发基于U-Net的肝脏分割系统的过程中，代码优化是提高模型性能和减少计算资源消耗的关键, 以下是在代码层面采取的一些优化措施：

**3.1.1 并行计算和GPU加速**

为了充分利用GPU的并行计算能力，对模型的关键部分进行了优化，以确保所有计算密集型任务都能在GPU上执行。这包括卷积操作、矩阵乘法和激活函数等。通过使用cuda()方法，将模型和数据移动到GPU上，能够显著加快训练的速度。

****

图14 选择GPU

**3.1.2 内存管理**

模型在训练过程中会消耗大量内存。通过优化数据加载和处理流程，减少了内存的使用。使用torch.utils.data.DataLoader的num\_workers参数来并行加载数据，减少了主进程的内存负担。

****

图15 并行加载数据

1. **实验论证**

**4.1 实验设置**

实验中使用的数据集包含肝脏的医学图像及其对应的分割标注。其中训练集包含500张原始图像和标签，用于模型的训练；测试集包含31张原始图像和标签，用于评估模型的性能。

**4.2 实验结果**

**4.2.1 训练损失与验证损失**

训练过程中，记录了每个epoch的训练损失和验证损失，用于监控模型的学习进度和过拟合情况。随着训练的进行，训练损失逐渐降低，验证损失也呈现下降趋势，但下降速度逐渐放缓，如图为100个epoch下的损失变化。

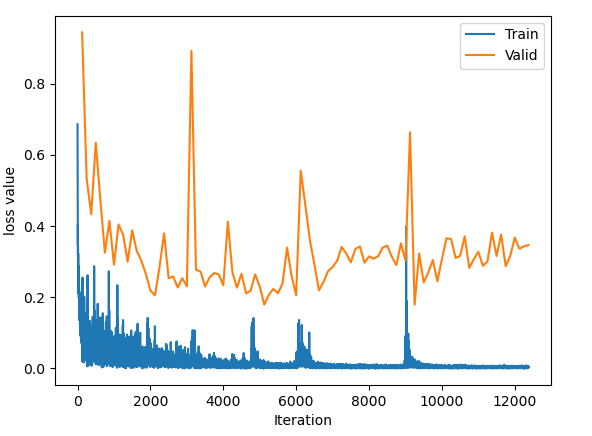


图16 损失变化图

**4.2.2 模型性能评估**

在验证集上，使用Dice系数[2]来评估模型的分割性能。Dice系数是衡量两个样本相似度的统计工具，值越接近1表示分割结果与真实标注越接近，Dice系数的计算公式如下：

在本任务中，X和Y分别是标签和分割图像，实验中计算了所有测试样本的Dice系数，并计算平均值以评估整体性能。可以看到在测试集中，Dice均值能达到0.881，分割结果于真实标注十分接近，说明模型分割结果较好。

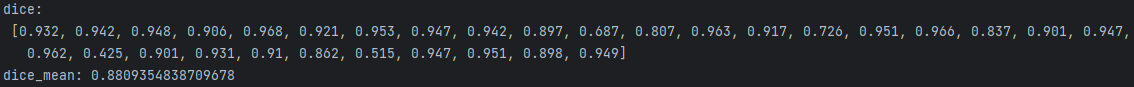


图17 dice均值

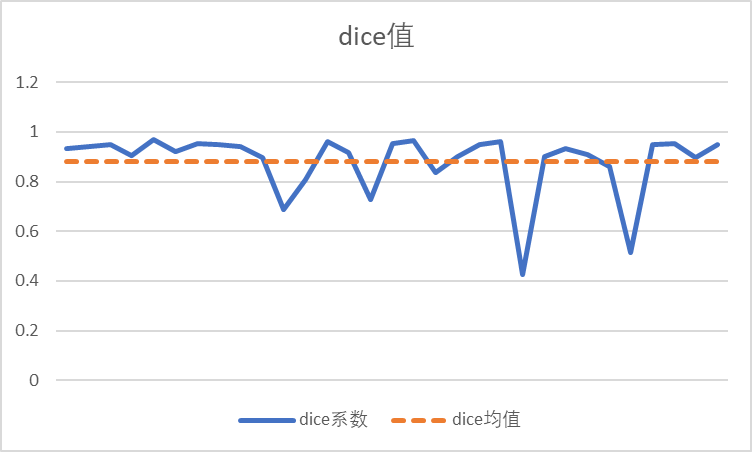


图18 图示化dice值

**4.2.3分割结果保存**

将分割结果保存在文件夹中，对于每个预测量分别保存其原始图像、标签以及分割结果。



图19 分割结果

**4.2.4 图形化窗口显示**

为了直观展示模型的分割效果，编写了GUI的图形化窗口，可以传入原始图像和标签，程序使用训练好的模型对原始图像进行预测，并显示其分割结果和Dice值。



图20 GUI部分

随机选取了几个样本，使用训练好的模型进行分割，并展示了原始图像、标签、分割结果以及计算得到的Dice值。观测可视化结果并结合Dice值可知，模型能准确地分割出肝脏区域，与真实标注具有较高的一致性。

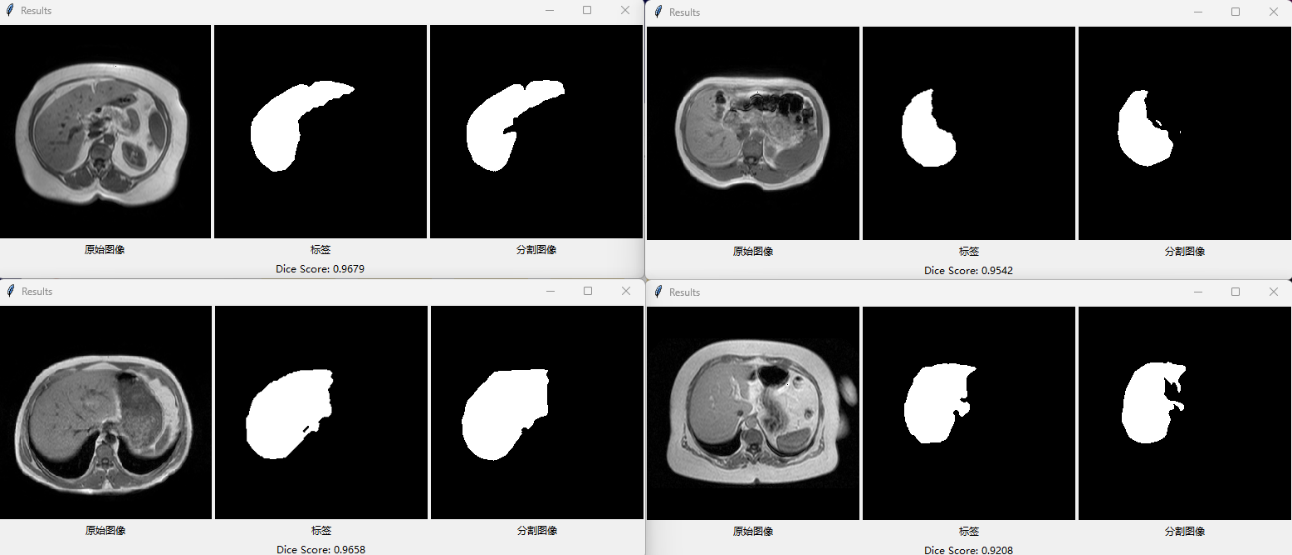


图21 分割结果举例

**【结论】**

本次实验实现了一个基于U-Net架构的肝脏分割系统，通过U-Net模型，实现了对肝脏区域的高精度分割，能有效地辅助了医生进行疾病诊断和治疗决策。实验结果表明，系统在肝脏分割任务上表现出了优异的性能，不仅在视觉上能够准确地识别和分割肝脏，而且在量化评估指标Dice系数上也达到了较好的水平。

在实验过程中，发现数据预处理、模型架构设计和训练策略对最终的分割效果有着显著的影响。合理的数据增强和归一化处理可以提高模型的泛化能力。通过GUI界面，提供了一个直观的交互平台，使得非专业用户也能方便地使用系统进行肝脏分割。

然而，受限于数据集的大小，本次实验的模型仍有待进一步优化，模型泛化能力不足，对实际图像的预测仍然有待加强。需要使用更多的医疗CT图像进行训练，并进一步联系医院进行验证。

【参考文献】

[1] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI).

[2] Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI).

【附录】

1. 字数统计

正文共4186个字，页数为10页

1. 补充说明 （可选）

（1）实验使用的库文件以及python版本如下：

|  |
| --- |
| Python = 3.10  opencv-python  pillow == 10.4.0  cuda == 12.1  pytorch == 2.4.0  torchvision == 0.19.0  matplotlib == 3.8.4  numpy == 1.26.4 |

（2）可执行程序gui.exe因为要读取模型，打开时间较长，双击后无反应请等待约1min。

（3） 代码文件以及数据文件（不包含训练模型和exe文件）约25MB，已打包上传到BlackBoard；全部文件（包含训练模型和exe文件）约2GB，上传至百度网盘：

通过百度网盘分享的文件：2022150221\_何泽锋\_Unet.zip

链接：https://pan.baidu.com/s/1VGLQnKOy-X\_Sl80Me7rPJw

提取码：5d3o