

深度学习与大数据智能课程

项目报告

****

**2020至2021学年第二学期**

论文题目： 基于深度卷积网络的肝脏分割

学生姓名： 20185570 姚语涵（第九组）

20183749 项怡雯（第九组）

20186290 薛艺瑶（第九组）

20181674 钟宏典（第九组）

学院： 计算机学院

任课教师： 文静

重庆大学计算机学院

**《深度学习》课程项目报告**

**目录**

[摘要 3](#_Toc75112247)

[关键词 3](#_Toc75112248)

[1引言 4](#_Toc75112249)

[2相关工作 4](#_Toc75112250)

[3数据 5](#_Toc75112251)

[3.1数据集介绍 5](#_Toc75112252)

[3.2数据预处理 6](#_Toc75112253)

[3.2.1 图像裁剪 7](#_Toc75112254)

[3.2.2 标签融合 7](#_Toc75112255)

[3.2.3 图像增强 7](#_Toc75112256)

[4系统模型 7](#_Toc75112257)

[4.1 深度神经网络整体结构 8](#_Toc75112258)

[4.1.1 U-Net网络 8](#_Toc75112259)

[4.1.2 DC-UNet网络 8](#_Toc75112260)

[4.1.3 UNet++网络 9](#_Toc75112261)

[4.1.4 基于ResNet的UNet++ 11](#_Toc75112262)

[4.2 深度神经网络结构详细解析 11](#_Toc75112263)

[4.2.1 卷积与池化 11](#_Toc75112264)

[4.2.2激活函数 12](#_Toc75112265)

[4.2.3损失函数 12](#_Toc75112266)

[4.2.4 空洞卷积 13](#_Toc75112267)

[4.2.5 残差网络 14](#_Toc75112268)

[4.3 最终模型架构 15](#_Toc75112269)

[4.4 模型对比结果 15](#_Toc75112270)

[5实验与分析 16](#_Toc75112271)

[5.1 数据集与评估标准 16](#_Toc75112272)

[5.2 实施细节 16](#_Toc75112273)

[5.3 评估模型 17](#_Toc75112274)

[5.4 分割测试可视化展示 20](#_Toc75112275)

[6结论 21](#_Toc75112276)

[参考文献 22](#_Toc75112277)

# 摘要

近年来，深度学习方法在医学图像分割领域取得了重大进展。采用编码-解码结构的深度学习网络U-Net在分割问题上获得了广泛的应用。UNet++在U-Net基础上进行改进，设计了具有嵌套和密集跳过连接的体系结构，可以结合多尺度特征实现精确分割。UNet++的下采样骨干网络仍是基于普通的CBR(Conv+BN+ReLU)模块堆叠的，将学习更强的ResNet作为UNet++的骨干网络，可以改进得基于ResNet的UNet++模型。

我们构建了U-Net以及基于ResNet的UNet++模型，并在Medical Segmentation Decathlon Challenge的肝脏分割数据集评估和测试了模型。实验结果表明，两种模型都具有不错的效果，而基于ResNet的UNet++模型性能更好。此外，我们将Focal Loss和Dice Loss结合，设计了一个混合损失函数，从而获得更准确的分割结果。

我们还将U-Net与空洞卷积结合，利用空洞卷积代替常规的卷积操作以扩大卷积感受野，构建并训练了DC-UNet。由于资源限制，我们没能测试该模型的分割效果，这可以作为未来工作的方向。

我们的代码公开在https://github.com/refrain-x/deeplearning

# 关键词：U-Net，UNet++，ResNet，DC-UNet，语义分割，医学影像

# 1 引言

人体器官检测和癌症细胞检测是人工智能在影像医学和病理方向的重要应用。该技术主要是应用了目前的深度学习技术。深度学习技术在医学图像分割领域具有非常重要的意义，能够实现自动用机器对人体器官进行识别和检测。因此，在深度学习方面，如何合理构建一个能够准确识别人体器官的框架成为了前沿研究者们研究的核心问题，得到了非常多的关注与应用。

本项目的数据是来自“Medical Segmentation Decathlon Challenge”的肝脏分割部分，分别为标签文件和图像文件，图像以3D方式进行展示。为了能够准确识别图像中的肝脏，进行肝脏分割，我们首先对数据集进行了数据预处理，将3D图像转换为2D图像，对于图像进行翻转切割等减小模型训练的代价，同时进行图像增强，提高图像训练的精度。在构建训练框架模块，选用了基础的U-Net框架，在其基础上添加了具有嵌套和密集跳过连接的体系结构，可以结合多尺度特征实现精确分割的UNet++，同时在UNet++的基础上添加残差网络，提升训练精度，形成我们的第一个训练框架——基于ResNet的UNet++。之后，我们了解到了空洞卷积在U-Net的方面的应用，我们尝试了在基础的U-Net架构上添加空洞卷积，并与第一个训练框架的训练效果进行对比。但是由于考虑到硬件配置和资源不足的问题，对于将U-Net与空洞卷积结合的DC-Unet我们仅训练了一部分数据，未对所有的数据进行模型训练，但是模型本身已经建立完成。因此，我们主要对比的数据是比较基础的U-Net和基于ResNet的UNet++的精度，可以发现后者的准确度可以达到0.911，训练损失仅0.1390，验证损失为0.1982，Dice系数可以达到93.10%，而前者的Dice系数仅有84.92%。因此，最终我们得到的精度最高的网络是基于ResNet的UNet++，其Dice Loss可以达到90%以上，模型准确度较高，损失较低，在肝脏识别模块中体现出了良好的效果。

下一步研究我们将从三方面展开尝试，首先是训练更大量的数据，这对提升模型的识别精度具有非常重大的意义，其次可以增加训练的epoch数，从50增加至100，观察模型精度的变化，最后在网络方面，我们可以尝试利用如MobileNet的网络进行构建新的模型或者加入Attention模块，从各个方面提升模型的精度。

# 2相关工作

目前国际上主流用于器官检测的模型有深度神经网络CNN等[1]，但深度神经网络需要专门的架构和训练方案修改才能达到竞争性能，大都是针对某一特定任务（如心脏分割）进行研究的，在这里我们挑选了适应性更好的U-Net进行模型的训练。

本次项目中我们小组所设计的模型使用到了U-Net结合空洞卷积、基于ResNet的UNet++两种网络结构来实现。 接下来将分别介绍两个模型中我们自己所设计的方法所引用的其他论文模型以及在此基础上的改进：

* 第一个模型：U-Net结合空洞卷积

U-Net是在医学图像处理方面非常著名的框架。U-Net是比较早的使用全卷积网络进行语义分割的算法之一，论文中使用包含压缩路径和扩展路径的对称U形结构在当时非常具有创新性，且一定程度上影响了后面若干个分割网络的设计[2]。而Dilated Residual Networks这篇文献创造性地提出了空洞卷积Dilated Convolution这个概念，它的有点在于内部数据结构的保留和避免了使用down-sampling这样的特性[4]。在我们框架的应用中，我们选择在不改变特征图的尺寸、不增加参数的情况下，加大感受野（Receptive Field）。

我们小组所做的工作便是将U-Net和空洞卷积结合DC-Unet，在下采样最后两层取消池化层，使用空洞卷积来完成肝脏图像的分割。

* 第二个模型：基于ResNet的UNet++模型

UNet++网络是U-Net网络的优化版，将原来空心的UNet++填满了。它的优势是可以抓取不同层次的特征，将它们通过特征叠加的方式整合不同层次的特征，或者说不同大小的感受野，它对于大小不一的目标对象的敏感度是不同的，比如，感受野大的特征，可以很容易的识别出大物体的，但是在实际分割中，大物体边缘信息和小物体本身是很容易被深层网络一次次的降采样和一次次升采样给弄丢的，这个时候就可能需要感受野小的特征来帮助。总的来看，UNet++是在U-Net的基础上改进了跳跃连接，使用了密集跳跃连接[3]。

其次是Deep Residual Learning for Image Recognition。阅读了文献后[5]，我们可以了解到ResNet是结构上看ResNet是正向的一个skip connection (shortcut)，通过shortcut实现恒等映射，能够解决深层网络效果反而下降和梯度消失的问题。

我们小组的工作便是将ResNet结合到UNet++中，将residual block添加到U-Net++中。并在此基础上加入了深度监督、在下采样的backbone使用ResNet。

# 3数据

## 3.1 数据集介绍

我们的项目为肝脏分割项目，数据是肝脏医学图像，格式为nii.gz，数据集为训练样本和标签各130个。数据集来自“Medical Segmentation Decathlon Challenge”的肝脏分割部分，数据集可从官网下载 (<http://medicaldecathlon.com/>)。官方数据集包含训练集(原始图像和标签)，测试集(仅有原始图像)，由于测试集没有对应标签，所以本次项目仅使用训练集。训练集将被再次划分为训练集、验证集和测试集。我们的训练集、验证集按照8:2划分，在所有数据集中，单独划分10个样本和标签为测试集。



图1：训练集、验证集、测试集的划分

可以看到样本中的图像都是由图2所示的3D图像进行展示，利用python的3D转换，我们可以将样本中的nii.gz图像用立体的形式表现出来。但是在我们的模型的处理中，将会主要应用2D图像对肝脏位置进行区分和识别。

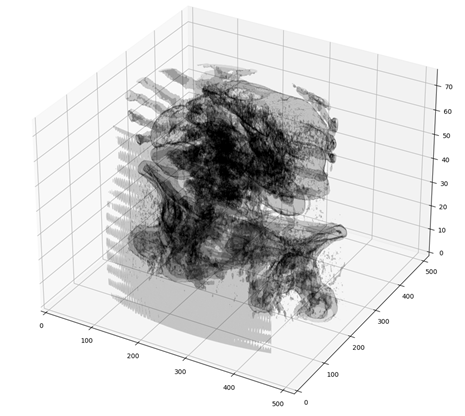


图2：肝脏3d图像

单个数据样本的维度：(slice, 512, 512) ，其中slice是样本通道数，因为肝脏大小的不同，使得每个样本具有不同通道数。

图3是通过python的simpleITK包对肝脏做切片后所呈现的图像。

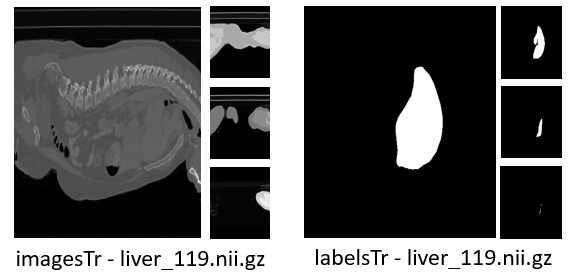


图3：119号肝脏图像及其标记

每一个nii.gz内部的样本平面图如上图所以。一般一个nii.gz样本中含30-120张2D图像。每个标本的样本又分为image和label样本。观察可以得到，标签样本LabelsTr 含有三个类别，分别是背景、肝脏、肿瘤，用像素值区分。

我们在该标签样本预处理时做二分类，将肿瘤视做肝脏的一部分，这是因为如果分为3类来训练，肿瘤太小会导致最后效果不佳。

## 3.2数据预处理

可以看到，我们总体的数据预处理过程如图4所示，主要将数据从基础的3维图像进行预处理和图像增强，之后将图像转换为2维的PNG格式，最后进行改变图像尺寸、剪切图像、翻转图像、扭曲图像等数据增强，之后将得到的PNG图片输入模型。

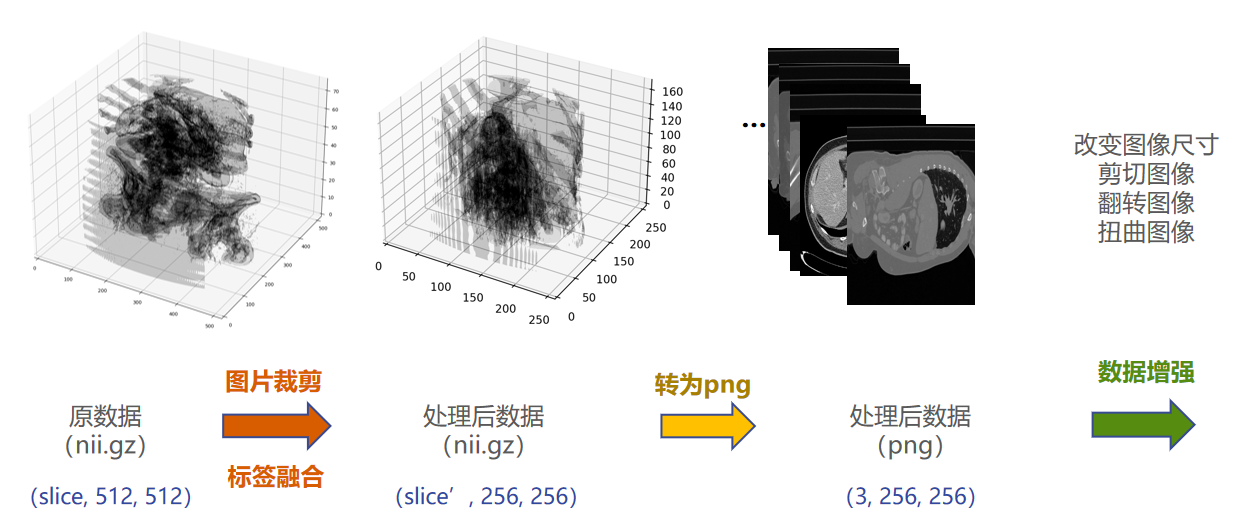


图4：数据预处理过程

具体的数据预处理手段有图像裁剪、标签融合、图像增强等。

### 3.2.1 图像裁剪

裁剪方式：在x、y横纵两个维度上寻找标签样本首个和最后一个像素出现的位置，并取首个像素出现位置的前x张slice以及最后一个像素出现的位置的后x张slice间的所有slice作为新的训练和标签样本，并将其存在fix\_list中。

降采样：使用scipy.ndimage.zoom缩小图像。

### 3.2.2 标签融合

对于标签样本，有背景、肝脏、肿瘤三种不同的像素值，我们将值大于0，也就是非背景的像素值全置1，将肝脏与肿瘤标签融为一个，只做二分类。

肝脏肿瘤融合：因为如果是三分类，肿瘤的大小相对于背景和肝脏来说太小。三分类最后效果很差，故考虑二分类。

### 3.2.3 图像增强

分别采用了翻转图像、改变图像尺寸大小和剪切图像、扭曲图像的技术来增强图像。

* 翻转图像：将图像上下翻转。
* 改变图像尺寸大小和剪切图像：将图像尺寸由512×512改变为256×256，仅留下图片含肝脏的重要部分，可以提高模型的训练效率。
* 扭曲图像：讲述集中对像素点实现三角函数变换,对图像做扭曲变化。

# 4系统模型

本部分我们将详细介绍基于深度卷积网络的肝脏分割的四种方法——U-Net网络模型、UNet++网络模型，Res-UNet++(UNet++与残差网络结合)的模型，DC-UNet（UNet与空洞卷积结合）的模型。第二小节将总体介绍四个网络模型的整体网络架构，第一小节将对网络结构各细节的作用及灵感来源进行详细介绍。

## 4.1 深度神经网络整体结构

### 4.1.1 U-Net网络

我们参考了《U-Net: Convolutional Networks for Biomedical Image Segmentation》[2]中提到的U-Net网络作为基础网络模型来实现。它的结构如图5所示：

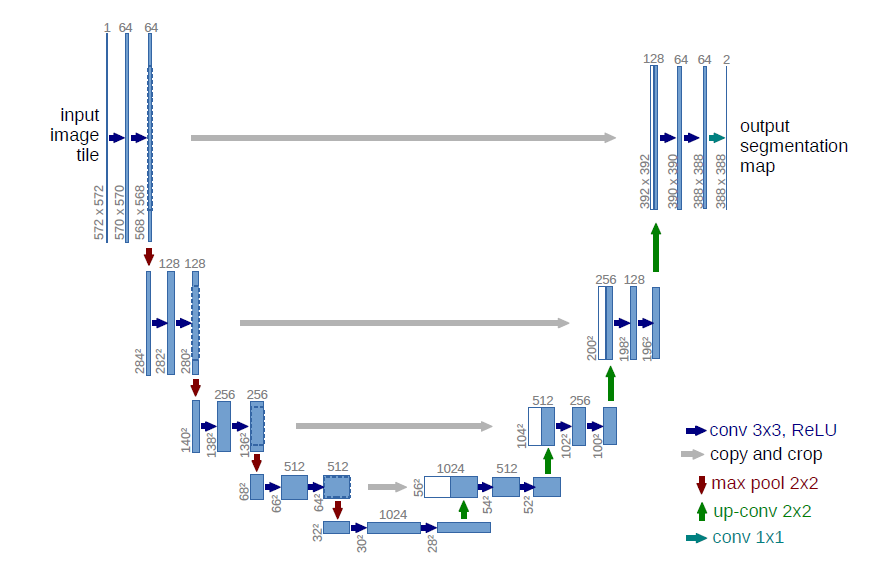


图5：U-Net架构

U-Net网络可以分为Encoder，Decoder和skip connection三个部分，主要用到了卷积，转置卷积，池化，跳跃连接，激活函数等技术。每一个蓝色块表示一个多通道特征图，特征图的通道数标记在顶部，X-Y尺寸设置在块的左下边缘。不同颜色的箭头代表不同的操作。图中左半部分是收缩路径，即Encoder模块；右半部分是扩展路径，即Decoder模块。

在Encoder部分由卷积操作和下采样操作组成，所用的卷积结构统一为 3×3 的卷积核，padding设置为1，striding设置为1。在Decoder阶段有卷积和上采样操作，上采样使用的方法是bilinear双线性插值。

每经过一次上采样通道数都会减半，再与收缩路径对应的特征图进行拼接。这就是skip-connection操作，它将浅层特征和深层特征拼接在一起。拼接之前需要进行crop，因为两者的尺寸并不相同，这主要是由valid卷积造成的。最后一层使用1×1大小的卷积核，将通道数降低至特定的数量（像素点级别数量）。

对于输入的大小，为了使得输出的分割图无缝拼接，选择输入块的大小是很重要的，以便所有的2×2的池化层都可以应用于偶数的x层和y层。在我们的实验中，输入处理后的图像的尺寸为256×256。

### 4.1.2 DC-UNet网络

仅仅用U-Net网络实现的分割效果并不理想，所以在U-Net基础上加入了空洞卷积。灵感来源于空洞卷积的特性。图像分割的输出结果是端到端的像素级分类标签，要求输出与输入具有相同的尺寸大小，在编码阶段进行多次卷积与池化操作后，输出的尺寸越来越小，所以在解码阶段需要使用上采样手段将尺寸还原到原始输入的大小，上采样一般采用转置卷积操作，之前的池化操作使得每个像素预测都能看到较大的感受野信息。因此在经典的图像分割网络模型全卷积神经网络中有两个关键，一个是通过池化操作减小图像尺寸，另一个是使用上采样还原图像尺寸。在尺寸变化的过程中，提取出了图像的语义信息，但是在缩放过程难免会有不少的信息损失，而空洞卷积是一种可以不通过池化缩小图像尺寸也能够具有更大的感受野获得更多的信息的操作，因此引入了空洞卷积。有关空洞卷积的知识将会在第二小节进行详细介绍。

DC-UNet结构框架如图6所示，主要用到了卷积，转置卷积，空洞卷积，池化，跳跃连接，激活函数等技术。其中总共有十八个卷积层，包栝六个普通卷积，六个转置卷积、六个空洞卷积，以及两种激活函数 Relu 和 Sigmoid；采用了两种合并技术：add和concat；加入了最大池化，以增强模型的平移旋转不变形。

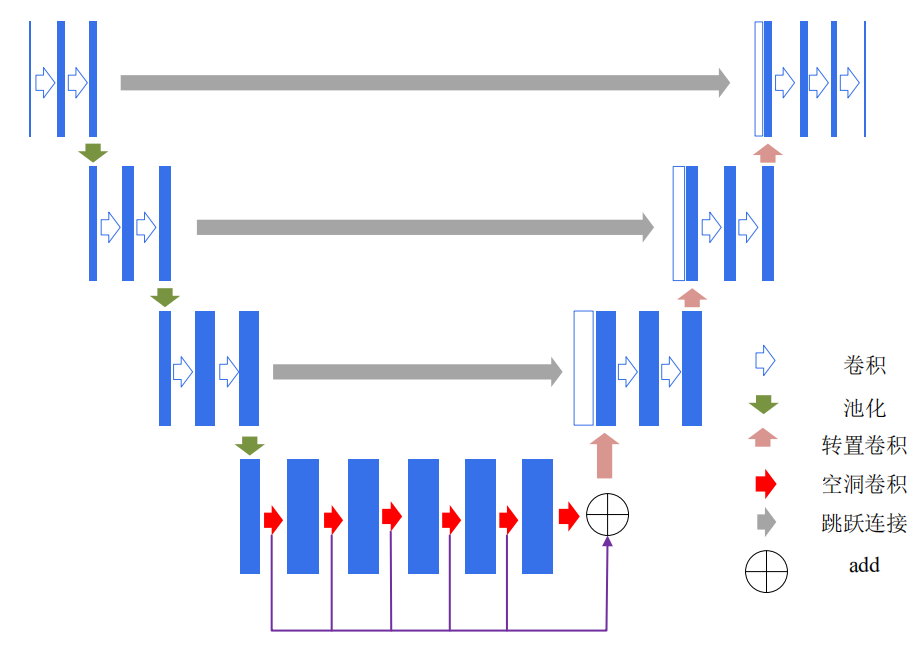


图6：DC-UNet架构

DC-UNet 网络分为四个阶段：编码阶段、特征融合阶段、解码阶段、像素分类阶段。在编码阶段采用的是类似 VGG 的卷积块。接下来进行了一系列的空洞卷积。每个空洞卷积都具有不同的空洞大小，分别采集到了图像中不同大小块的信息。然后将其叠加得到更为丰富的组合信息，有利于后续的训练。由于图像分割是一个输入等于输出的模型，所以之后进行了三次上采样，采用的是转置卷积操作。然后通过跳跃连接将每一层转置卷积的输入合并网络中对应位置的下采样的输出。最后的分类没有采用传统的全连接层，而是同样用了卷积层。一是减少了网络的参数；二是可以支持任意大小的图片输入；三是可以取得和全连接差不多的效果。

### 4.1.3 UNet++网络

为了进一步优化U-Net网络，参考论文《UNet++:A Nested U-Net Architecture for Medical Image Segmentation》的方法实现了UNet++网络，利用多尺度特征融合来优化U-Net网络。UNet++网络结构如图7：

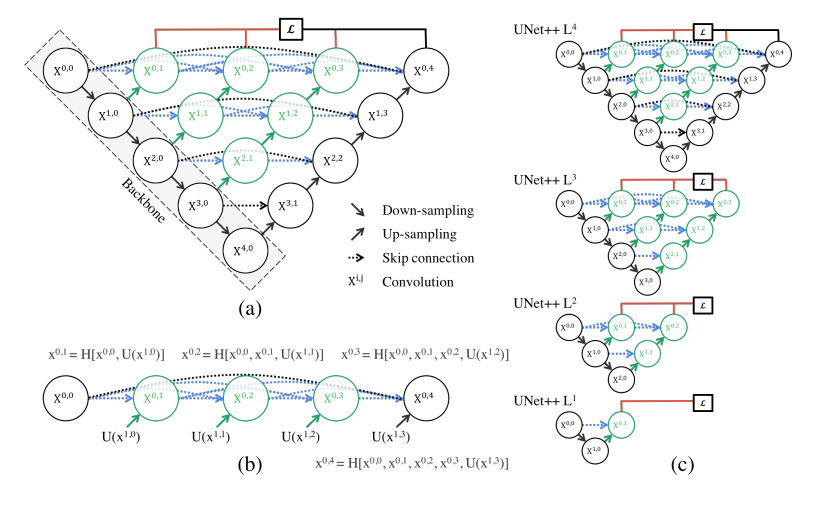


图7：U-Net++网络架构

UNet++ 在原始的U-Net上主要改进了3个方面：

1. 重新设计的跳跃路径（显示为绿色） 以弥补编码器和解码器子路径之间的语义差别。U-Net采用跳跃连接，直接连接编码器和解码器之间的特征映射，导致把语义上不相似的特征映射相融合。然而，在UNet++中，相同dense block的前一个卷积层的输出与较低层dense block对应的上采样输出进行融合。这使得已编码特征的语义级别更接近于等待在解码器中的特征映射的语义级别，因此，当接收到语义上相似的特征映射时，优化更容易。

2. 密集跳跃连接（显示为蓝色） 实现了编码器和解码器之间的跳跃路径。这些Dense blocks是受到DenseNet的启发，目的是提高分割精度和改善梯度流。密集跳跃连接确保所有先验特征图都被累积，并通过每个跳跃路径上的dense卷积块而到达当前节点。这将在多个语义级别生成完整分辨率的特征映射。

3. 深度监督（显示为红色） 通过修剪模型来调整模型的复杂性，在速度(推理时间)和性能之间实现平衡。

关于深度监督的设计源于在反向传播的时候,如果只用最右边的一个loss来做的话, 中间部分会收不到过来的梯度，导致无法训练, 为了解决中间部分接收不到梯度的问题，加入了深度监督。如下图所示,具体的实现操作就是在图中X0,1、X0,2、X0,3、X0,4后面加一个1×1的卷积核，相当于去监督每个层次，或者说监督每个分支的U-Net的输出。如图8所示：

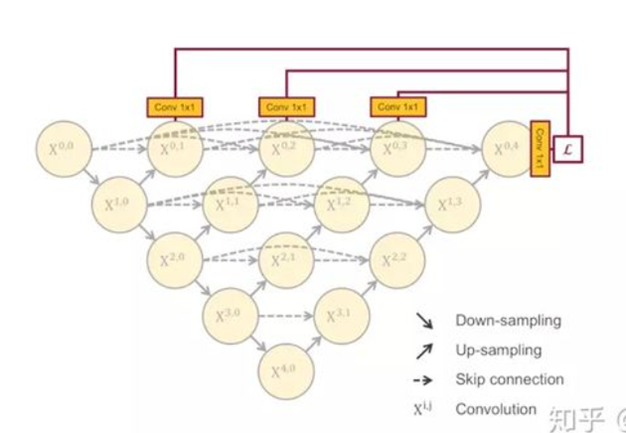


图8：加入深度监督的UNet++

并且，通过在训练过程中在各层子网络中加入深监督，UNet++可以实现剪枝，将每个剪完剩下的子网络根据深度命名为UNet++ L1，L2，L3，L4，加入了深度监督后，每个子网络的输出其实已经是图像的分割结果了，所以如果小的子网络的输出结果足够好，就可以剪掉多余的部分。

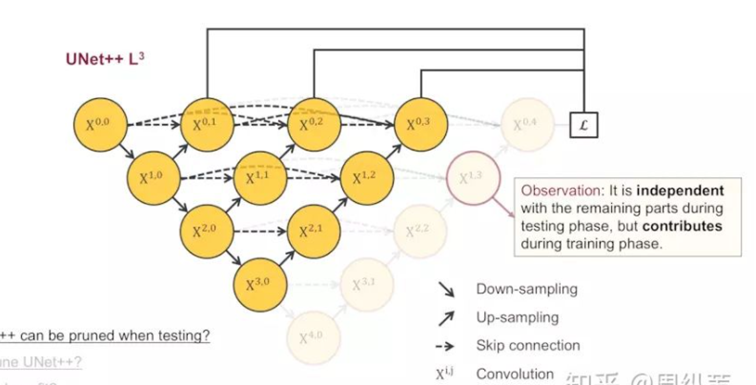


图9：新的UNet++架构

### 4.1.4 基于ResNet的UNet++

我们在U-Net网络做下采样时BackNone是由普通的CBR模块(Conv+BN+ReLU)堆叠而成，为了进一步优化UNet++，我们使用ResNet作为UNet++的backbone，UNet++可以看成多层U-Net叠加在一起，所以每一层都引入残差即可实现该模型。具体的残差结构将在4.2小节介绍。

## 4.2 深度神经网络结构详细解析

本节将从卷积与池化、激活函数、损失函数、空洞卷积及残差网络等几个方面，详细介绍Unet以及进阶网络的设计思想。

### 4.2.1 卷积与池化

在卷积神经网络模型中，卷积和池化是必不可少的操作。通过卷积核与输入图像进行卷积操作，可以提取图像的特征。提取出来的特征的好坏是由卷积核来决定的，卷积核的参数是通过训练学习得来的，而卷积核的大小则需要人工指定，卷积核通常是F×F的矩阵。一般情况下卷积核的大小与模型的深度有关，模型设计的深度越深则卷积核的尺寸越小，因为在输入图像尺寸相等的情况下，卷积核的尺寸越大，输出特征图的尺寸越小。我们的神经网络U-Net在编码阶段采用了 VGG 网络的结构，该过程分为三段卷积过程，每段卷积由两个卷积层和一个池化层组成。其中卷积核的大小设置成了3×3，移动步长为1，同时为了保证输出特征图与输入图像的尺寸相同，采用了补丁操作，在图像的每个边添加了一个像素的0值。

池化在卷积神经网络中的作用主要有三个：第一是使输入的特征维度变得更小，减少网络中的参数和计算的数量，达到控制过拟合的目的，同时限制网络的规模可以是模型更加容易训练；第二是使网络对于输入图像中的更小的变化、冗余和变换具有不变性；第三是使训练模型对图像具有最大程度上的尺度不变性，即对于图像中的物体，无论在图像中的哪个位置都可以检测出来。

我们实验中使用的是最大池化。最大池化是选取邻域内的最大值作为输出结果，如图10所示：

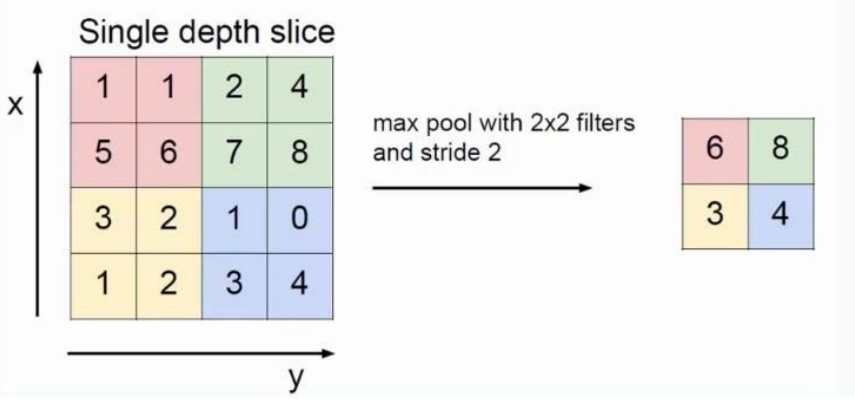


图10：最大池化

### 4.2.2激活函数

通常在神经网络的网络层之间使用激活函数，下一个网络层的输入是上一个

网络层的输出通过激活函数进行转换之后的结果。使用激活函数的目的是是神经

网络模型具有更强的表现力，能够描述各种非线性问题。

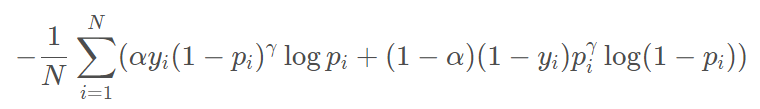
我们的实验在编码、特征融合以及解码阶段采用了 ReLU 函数作为激活函数。当输入信号的值小于0时，输出结果都是0；输入信号的值大于0时，输出则等于输入。

### 4.2.3损失函数

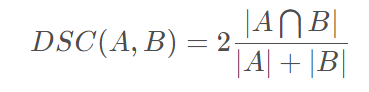
深度学习中的每一个的算法都有一个损失函数，损失函数是用来评价模型的

预测值与数据集的真值之间的差异的。输入是训练集的真值和模型得到的预测值，通过损失函数计算输出预测值与真值之间的差别，差别越趋近于0，说明模型的效果越好，反之则越差。损失函数的选择很大程度的影响着模型的好坏，只有能够真实地体现出预测值与真值差异的损失函数才能正确的反馈模型的质量，以便于做出相应的调整，训练处高质量的模型。

Focal loss的提出是在目标检测领域，为了解决正负样本比例严重失衡的问题。公式如下：



Dice loss 的提出是在V-Net中，其中的一段原因描述是在感兴趣的解剖结构仅占据扫描的非常小的区域，从而使学习过程陷入损失函数的局部最小值。所以要加大前景区域的权重。Dice 可以理解为是两个轮廓区域的相似程度，用A、B表示两个轮廓区域所包含的点集，定义为：



我们将Focal Loss和Dice Loss结合，设计了一个混合损失函数，从而获得更准确的分割结果。

### 4.2.4 空洞卷积

感受野是深度学习中的重要概念，指的是卷积神经网络结构中某个特征映射

到输入空间的区域大小。感受野的大小代表了提取的特征包含信息的多少，感受野越大包含的上下文信息越多。在图像分割中，提取的特征包含的信息越多则对当前像素点进行正确分类的可能性就越大。通过增加卷积层可以增大感受野，如图11所示。

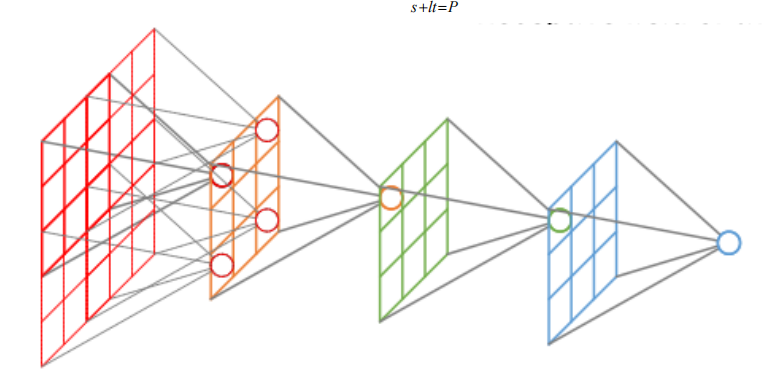
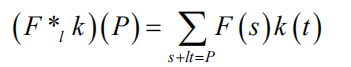


图11：空洞卷积结构

但是增加卷积层会增大网络的复杂度，降低训练速度；第一种解决方法是使用池化层，先降低图像的尺寸来增大感受野，然后对结果采用上采样恢复到原来的尺寸，但是在先减小再增大尺寸的过程中，存在不少的细节信息损失；另一种方法是使用空洞卷积，空洞卷积可以在不损失信息的情况下增大感受野。假设一个卷积神经网络所有的卷积核都是k×k的，卷积的步长为1，那么一个卷积核对应单元的感受野，即用来激活该单元的像素块的大小就是(k-1)×1+1 ，即感受野的大小与层数l成线性关系。这对于那些高分辨率的输入图像来说存在很大的限制，需要叠加更多的卷积层才能提取到高质量的特征。空洞卷积的提出就是解决此问题的。二维空间的空洞卷积的定义如下：



其中F是输入的二维信号（图像），s 是其定义域；k 是核函数，t 是其定义域；l 是空洞卷积的系数；P 是空洞卷积的定义域。

可以看到，与普通的卷积相比，卷积的条件从s+t=P变成了s+lt=P，即每次卷积核仅仅和图像F中l倍数位置的元素运算。图12展示了空洞卷积与感受野之间的关系。

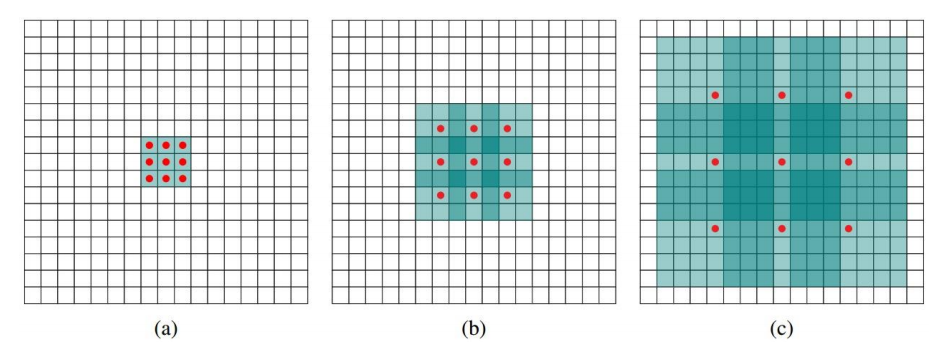


图12：空洞卷积与感受野的关系

卷积核均为3×3，(a)图是空洞卷积系数为1时的感受野，与普通的卷积操作一样；(b)图卷积系数为2的感受野，感受野是7×7的图像区域，只有9个标记为红色的点与卷积核发生了卷积操作，其余的点略过。可以看到虽然卷积核的尺寸只有3×3，但是这个它的感受野已经得到了极大地扩增，达到了7×7；(c)图是系数为4时的感受野，感受野是15×15的图像区域。与自身相比，当卷积核增大时，空洞卷积的感受野呈指数级增长；与传统卷积操作相比，在参数没有增加的情况下感受野增加了很多。这种特性很适合在图像分割方面使用，所以我

们将其加入到我们的图像分割模型中。在参数不增加的前提下，达到了很好的效果。

在DC-UNet网络中，在编码阶段结束后，我们对获取的特征分别进行系数为1、2、4、8、16、32 的空洞卷积，进一步提取不同尺度的特征，然后将不同尺度的特征进行融合，得到表现力更强的综合特征。

### 4.2.5 残差网络

浅层网络都是希望学习到一个恒等映射函数H(x)=x，其中=指的是用H(x)这个特征/函数来代表原始的x的信息，但随着网络的加深这个恒等映射变得越来越难以拟合。即是用BN这种技巧存在，在深度足够大的时候网络也会难以学习这个恒等映射关系。因此ResNet提出将网络设计为H(x)=F(x)+x，然后就可以转换为学习一个残差函数F(x)= H(x)-x，只要残差为0，就构成了一个恒等映射H(x)=x，并且相对于拟合恒等映射关系，拟合残差更容易。残差结构具体如图13所示：

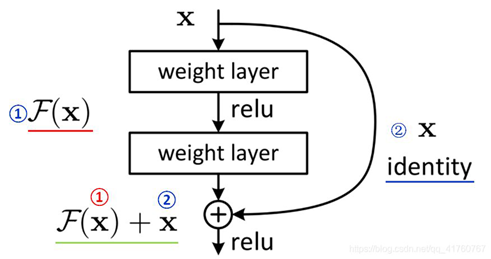


图13：残差结构

Identity mapping表示恒等映射，即将浅层网络的特征复制来和残差构成新的特征。其中恒等映射后面也被叫作跳跃连接。同时我们可以看到一种极端的情况是残差映射为0，残差模块就只剩下x，相当于什么也不做，但这至少不会带来精度损失。我们将该模型引入UNet++中作为其下采样的backbone。

## 4.3 最终模型架构

根据以上模型，我们最终形成了两个模型，模型的整体架构如图14所示。第一个模型为结合空洞卷积的U-Net框架，第二个模型为基于ResNet的UNet++。

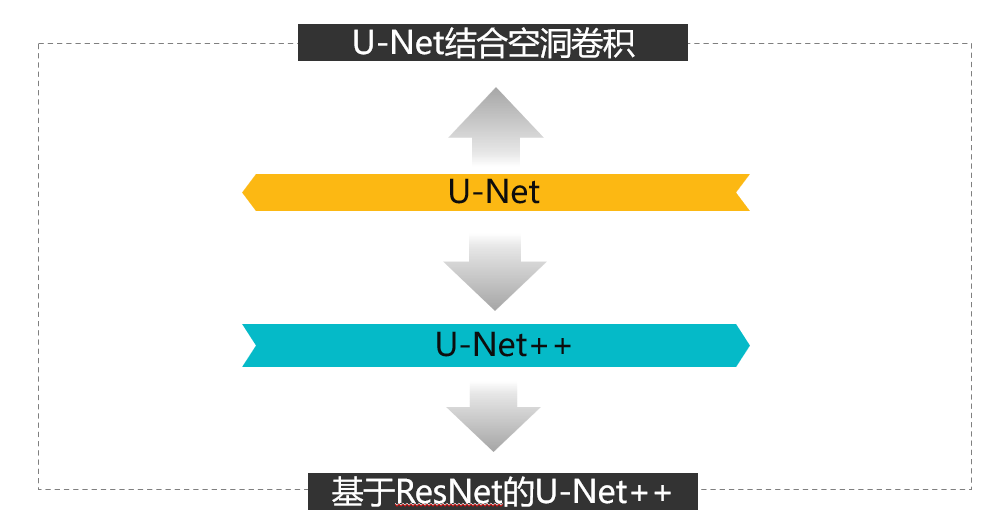


图14：最终模型架构

## 4.4 模型对比结果

如图15，可以看到使用不同的模型对于精确度的提升效果，Res-UNet++的精确度是比较高的，并且比普通的U-Net的精确度高了0.1左右。

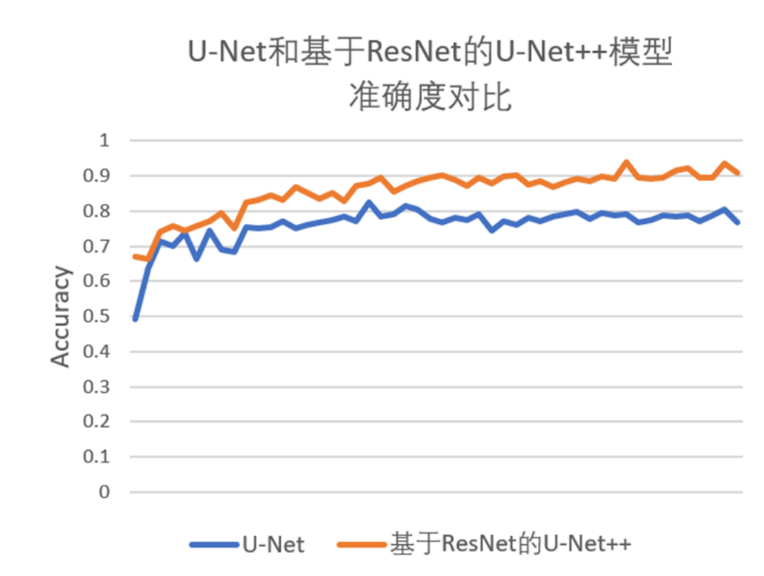
****

图15：U-Net和基于ResNet的UNet++模型准确度对比

对于模型的训练损失和测试损失来说，也可以看到基于ResNet的UNet++模型的损失远低于普通的U-Net的模型。具体的数据和模型优化效果将于Experiments中进行介绍。

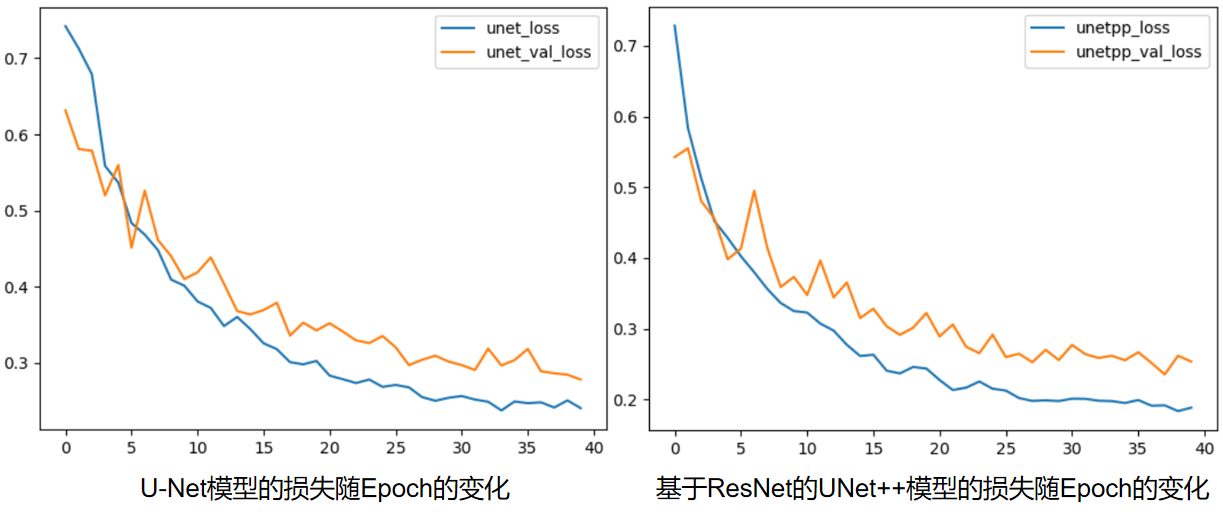


图16：U-Net和基于ResNet的UNet++模型训练损失与验证损失对比

# 5实验与分析

## 5.1 数据集与评估标准

我们在预处理后的Medical Segmentation Decathlon Challenge的肝脏分割数据集上进行了模型评估，采用训练损失和验证损失以及Dice系数来评估模型的性能。

## 5.2 实施细节

我们基于PyTorch实现了我们的模型。设备为华为云GPU: 2\*NVIDIA-V100NV32(32GB) | CPU: 16 核 128GB的Notebook开发环境。我们将数据集打乱，按照8:2:1划分训练集、验证集和测试集，其中各包含100个、20个和10个未分割样本和标签样本。我们的训练集、验证集和测试集交集均为空。

训练过程中，我们设置batch大小为2，输入为3 × 256 × 256的PNG图像，设置损失函数为Focal Loss和Dice Loss结合的混合损失函数。整个网络采用Adam算法进行更新，初始学习率设为10-2，根据训练次数调整学习机制，在每个Epoch衰减为原来的0.9，共训练50个Epoch。此外，为了防止出现过拟合，我们使用了早停策略，当训练损失下降而验证损失上升时，停止训练。

## 5.3 评估模型

我们分别比较了U-Net模型和基于ResNet的UNet++模型的损失随Epoch的变化。可以看到，在50个Epoch内，在两种模型上，不论是训练还是验证损失均呈现下降趋势，并没有出现过拟合的情况，模型性能稳步提升。

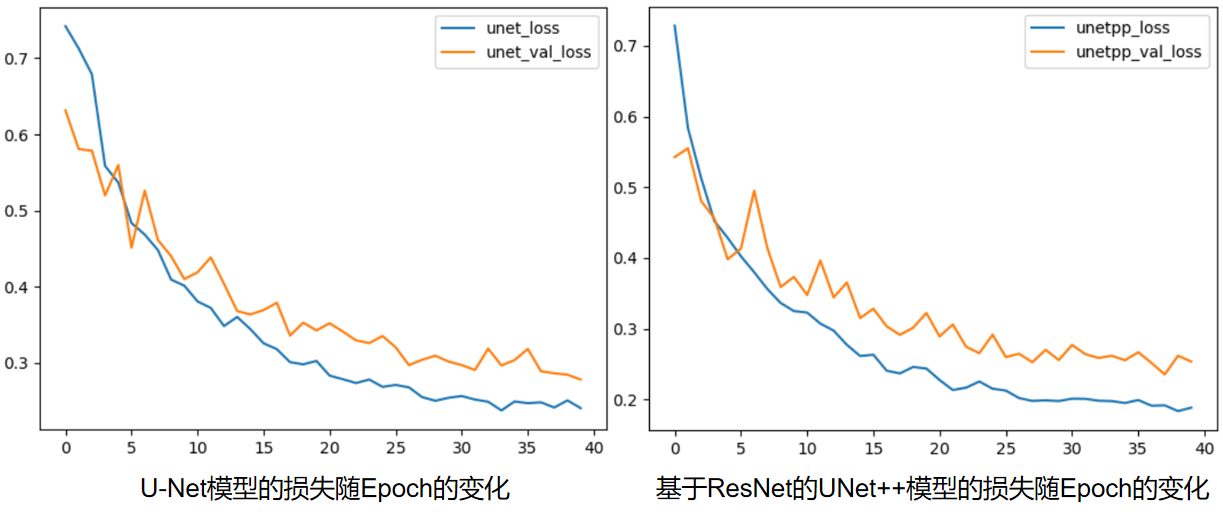


图17：U-Net模型和基于ResNet的UNet++模型的损失随Epoch的变化

通过对比两种模型在训练集和验证集上的损失随Epoch的变化，我们可以看出基于ResNet的UNet++模型的损失下降更快，且损失值更小，证明该模型比U-Net模型收敛速度快，同时模型的性能更好。

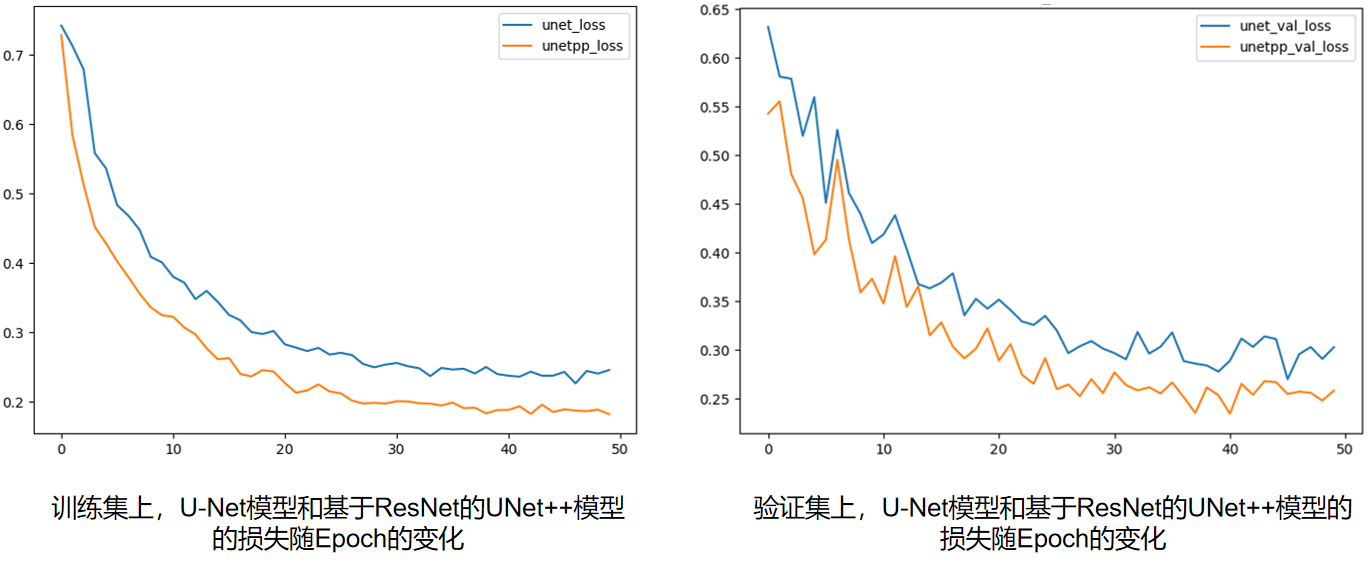


图18：对比U-Net模型和基于ResNet的UNet++模型的损失变化趋势

我们对于两种模型的准确度进行了对比，U-Net模型的准确度为0.845，基于ResNet的UNet++模型为0.911，可见基于ResNet的UNet++模型比起U-Net模型有显著的准确度提升。

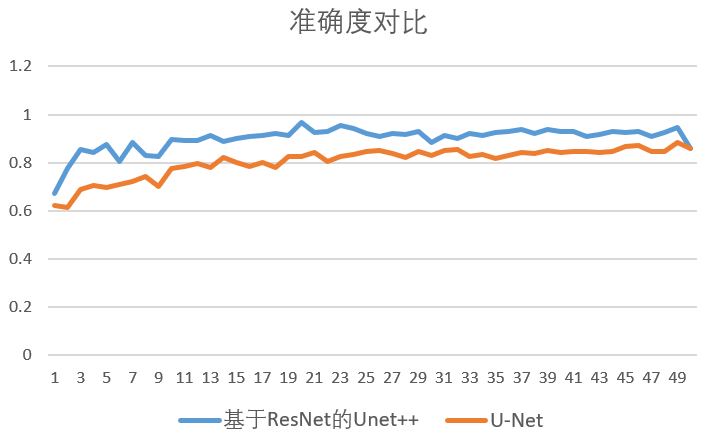


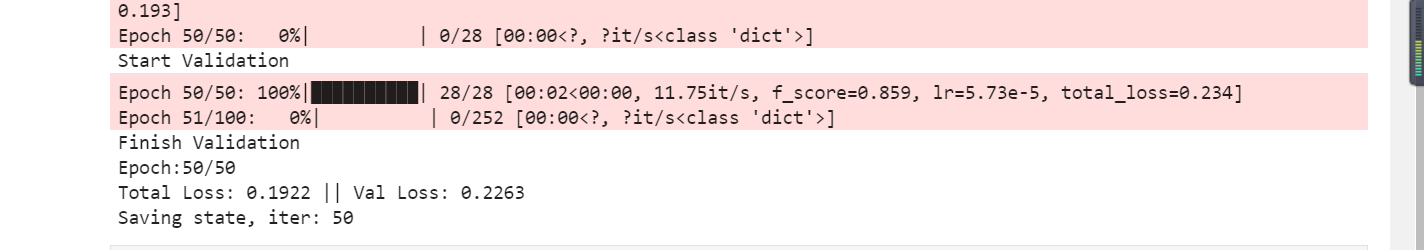
图19：基于ResNet的UNet++和基础U-Net模型的准确度对比

模型在每个训练epoch的具体损失和准确度如表1所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 基于ResNet的Unet++ | | | U-Net | | |
| train loss | value loss | accuracy | train loss | value loss | accuracy |
| 0.8141 | 0.5539 | 0.622 | 0.8219 | 0.6288 | 0.6204 |
| 0.7285 | 0.5428 | 0.7752 | 0.7422 | 0.6316 | 0.6128 |
| 0.584 | 0.5553 | 0.8526 | 0.713 | 0.5807 | 0.6896 |
| 0.5129 | 0.4807 | 0.8405 | 0.679 | 0.5786 | 0.7061 |
| 0.4522 | 0.4559 | 0.8771 | 0.5585 | 0.5198 | 0.6949 |
| 0.4286 | 0.3983 | 0.8044 | 0.5367 | 0.5597 | 0.7092 |
| 0.4026 | 0.4133 | 0.8856 | 0.4835 | 0.4512 | 0.7199 |
| 0.38 | 0.4953 | 0.8316 | 0.4684 | 0.5261 | 0.7439 |
| 0.3562 | 0.4139 | 0.8239 | 0.448 | 0.4613 | 0.7001 |
| 0.3365 | 0.3591 | 0.8958 | 0.4092 | 0.4399 | 0.7743 |
| 0.325 | 0.3733 | 0.8904 | 0.4012 | 0.4099 | 0.7819 |
| 0.3229 | 0.3478 | 0.8929 | 0.3805 | 0.4188 | 0.7964 |
| 0.3073 | 0.3965 | 0.9113 | 0.372 | 0.4384 | 0.7811 |
| 0.2974 | 0.3443 | 0.8896 | 0.3482 | 0.4037 | 0.8197 |
| 0.2773 | 0.3654 | 0.9004 | 0.3601 | 0.3678 | 0.8007 |
| 0.2615 | 0.3151 | 0.909 | 0.3441 | 0.3634 | 0.7843 |
| 0.2633 | 0.3284 | 0.9136 | 0.3255 | 0.3692 | 0.8007 |
| 0.2408 | 0.3035 | 0.923 | 0.3178 | 0.3788 | 0.7787 |
| 0.2368 | 0.2915 | 0.9126 | 0.3007 | 0.3357 | 0.8236 |
| 0.246 | 0.3014 | 0.9663 | 0.2979 | 0.3526 | 0.8273 |
| 0.2438 | 0.3223 | 0.9247 | 0.3023 | 0.3425 | 0.8441 |
| 0.2275 | 0.2892 | 0.9311 | 0.283 | 0.3518 | 0.8042 |
| 0.2134 | 0.3061 | 0.9546 | 0.2783 | 0.341 | 0.8234 |
| 0.2167 | 0.2746 | 0.9433 | 0.2733 | 0.3293 | 0.8342 |
| 0.2254 | 0.2655 | 0.9192 | 0.2779 | 0.3258 | 0.8471 |
| 0.2152 | 0.2917 | 0.9076 | 0.2683 | 0.3351 | 0.8508 |
| 0.2126 | 0.26 | 0.921 | 0.2708 | 0.3202 | 0.839 |
| 0.202 | 0.2647 | 0.9151 | 0.2676 | 0.2968 | 0.8213 |
| 0.198 | 0.2527 | 0.9307 | 0.2548 | 0.3039 | 0.8457 |
| 0.1988 | 0.2702 | 0.8843 | 0.25 | 0.3092 | 0.8296 |
| 0.1978 | 0.2558 | 0.9119 | 0.2539 | 0.3015 | 0.8503 |
| 0.2011 | 0.2771 | 0.8999 | 0.2562 | 0.2968 | 0.8531 |
| 0.2009 | 0.2641 | 0.9212 | 0.2516 | 0.2905 | 0.8238 |
| 0.1983 | 0.2587 | 0.9113 | 0.2487 | 0.3185 | 0.835 |
| 0.1977 | 0.2618 | 0.9243 | 0.2373 | 0.2964 | 0.8177 |
| 0.195 | 0.2555 | 0.9308 | 0.2491 | 0.3035 | 0.8314 |
| 0.1992 | 0.2668 | 0.9375 | 0.2468 | 0.3181 | 0.8422 |
| 0.1912 | 0.2516 | 0.9193 | 0.2479 | 0.2887 | 0.8361 |
| 0.1917 | 0.2354 | 0.936 | 0.2411 | 0.2861 | 0.8501 |
| 0.1836 | 0.2618 | 0.9294 | 0.2505 | 0.2842 | 0.8406 |
| 0.1884 | 0.2536 | 0.9302 | 0.2403 | 0.2779 | 0.8478 |
| 0.1886 | 0.2347 | 0.9082 | 0.2379 | 0.2891 | 0.8457 |
| 0.1938 | 0.2654 | 0.9157 | 0.2365 | 0.3118 | 0.8435 |
| 0.1829 | 0.2541 | 0.929 | 0.2437 | 0.3032 | 0.8458 |
| 0.1962 | 0.2681 | 0.9256 | 0.2379 | 0.314 | 0.8653 |
| 0.1856 | 0.2671 | 0.9288 | 0.238 | 0.3112 | 0.8722 |
| 0.1895 | 0.2549 | 0.9104 | 0.2434 | 0.2701 | 0.8456 |
| 0.1877 | 0.2574 | 0.927 | 0.2268 | 0.2958 | 0.846 |
| 0.1868 | 0.2561 | 0.944 | 0.2448 | 0.3031 | 0.8855 |
| 0.1891 | 0.2482 | 0.9091 | 0.241 | 0.2909 | 0.8596 |

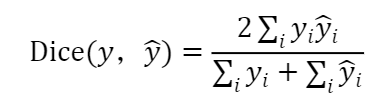
表1：基于ResNet的UNet++和基础U-Net模型在每个epoch的具体损失和准确度

由于资源限制，我们只取得了将U-Net与空洞卷积结合的DC-UNet的训练结果，其最终训练损失值为0.1922，验证损失值为0.2263。



图：将U-Net与空洞卷积结合的DC-UNet的训练结果

由于我们主要通过Dice系数来衡量模型的效果。标准的Dice系数由下面的公式计算得到：



其中代表网络的预测结果，而代表真值图。在本次项目中，最终要求得到的DICE系数需大于80%，而我们最终模型的Dice系数远远超过80%达到了93%。

由下图可以看到，基于ResNet的UNet++网络的Dice loss要远远高于基础的U-Net，在添加了ResNet和UNet++之后，U-Net的Dice loss由84.92%提升到了93.10%，增加了近10个百分点，模型的训练和识别肝脏的能力得到巨大的提升。

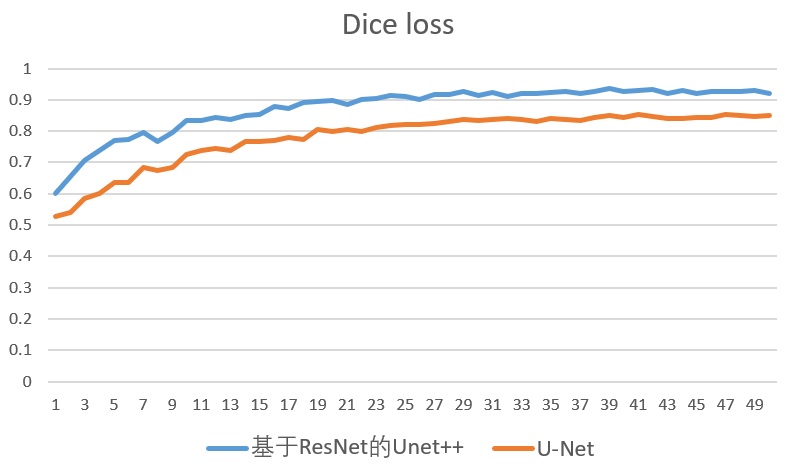


图20：基于ResNet的UNet++和基础U-Net模型的Dice系数对比

## 5.4 分割测试可视化展示

我们将训练好的模型在测试集上进行分割测试。图21展示了最终批量产生的肝脏分割效果。图22显示了不同网络的分割效果，定性地比较了这两种模型的预测结果之间的差异。与标签样本（即真值）比较，可以看出相对于U-Net，基于ResNet的UNet++模型可以得到更好的分割结果。

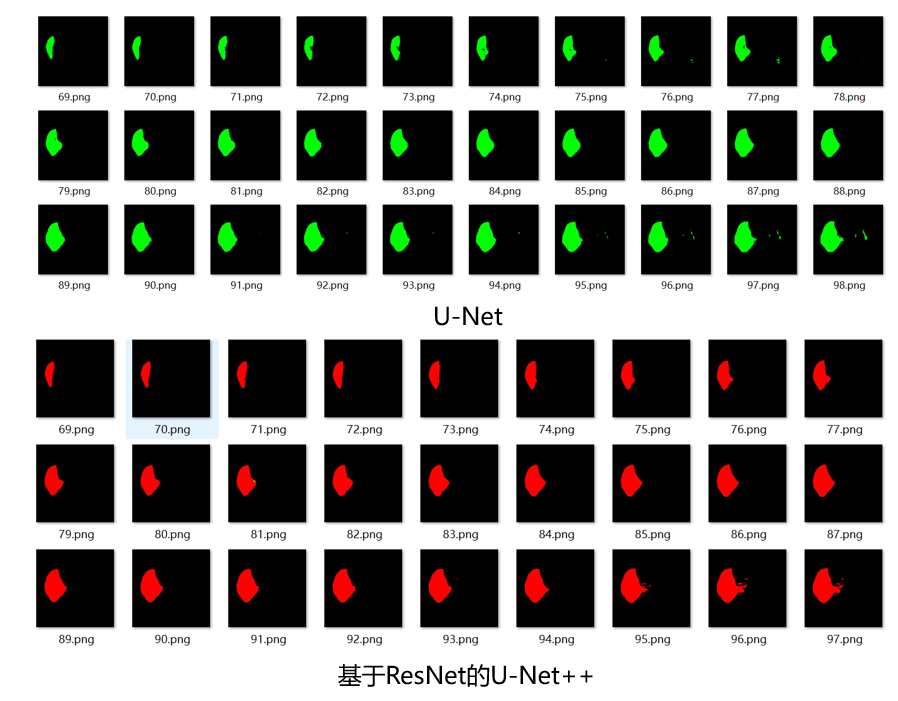


图21：U-Net模型和基于ResNet的UNet++模型的批量分割结果

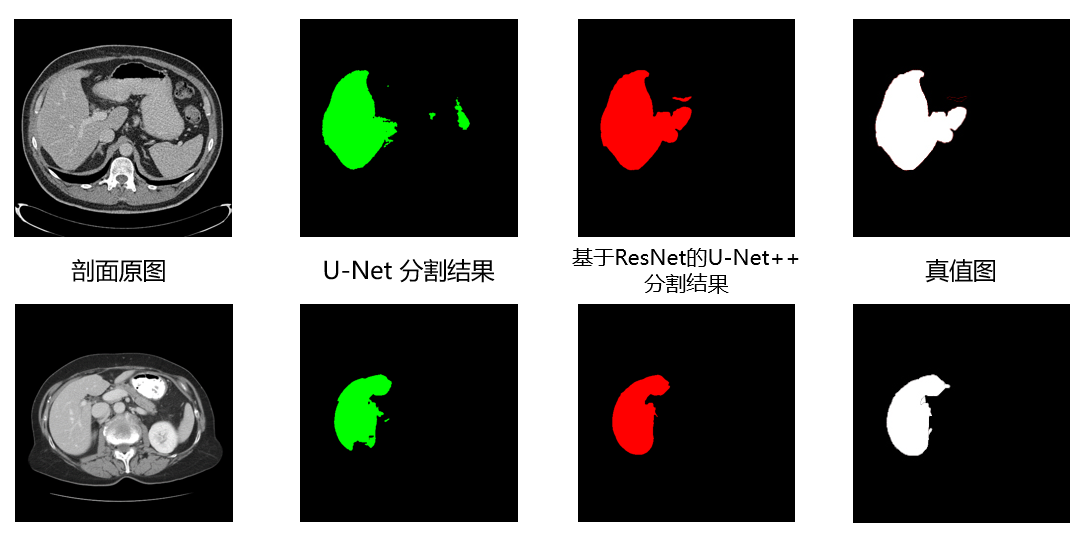


图22：U-Net模型和基于ResNet的UNet++模型的分割结果

# 6结论

我们构建了采用编码-解码结构的深度学习网络U-Net，以及在UNet++基础上做出改进，将其的骨干网络替换为学习能力更强的ResNet的于ResNet的UNet++模型。同时，我们将Focal Loss和Dice Loss结合，设计了一个混合损失函数，从而获得更准确的分割结果。我们在Medical Segmentation Decathlon Challenge的肝脏分割数据集评估和测试了模型，实验结果表明，两种模型都具有不错的效果，而对比U-Net，基于ResNet的UNet++模型性能更好，分割结果更准确。

我们还将U-Net与空洞卷积结合，利用空洞卷积代替常规的卷积操作以扩大卷积感受野，构建并训练了DC-UNet。由于资源限制，我们没能测试该模型的分割效果，这可以作为未来工作的方向。

我们的工作仍然具有改进空间。对于数据方面，其一，我们的数据集对于训练样本以万计数的深度学习课题而言显然过小，这也是阻碍我们模型性能提升的重要问题之一；其二，受制于设备，我们没有探索出最佳的Epoch数。对于网络结构而言，我们仅构建了U-Net网络和可获得性能提升的基于ResNet的UNet++模型。在骨干网络方面，可以尝试其他同样被证明对特征提取非常有效的网络，如MobileNet[6]；同时，还可以加入其他构件，如能够更好聚焦特征的Attention模块[7]；此外，还能尝试不同的卷积方式，如我们已经使用的空洞卷积，或是用3D卷积[8]等。基于我们对网络结构的进一步改进，我们还可以进行消融研究，以探究不同的网络结构的改进对于分割结果的影响。同时，我们可以在不同的肝脏数据集上测试我们模型的分割效果，以验证其泛化性能。

# 参考文献

[1] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).

[2] Ronneberger O , Fischer P , Brox T . U-Net: Convolutional Networks for Biomedical Image Segmentation[J]. Springer, Cham, 2015.

[3] Zhou Z , Siddiquee M , Tajbakhsh N , et al. UNet++: A Nested U-Net Architecture for Medical Image Segmentation[J]. 4th Deep Learning in Medical Image Analysis (DLMIA) Workshop, 2018.

[4] Yu F , Koltun V , Funkhouser T . Dilated Residual Networks[C]// IEEE Computer Society. IEEE Computer Society, 2017.

[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[6] Howard A G , Zhu M , Chen B , et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.

[7] Oktay O , Schlemper J , Folgoc L L , et al. Attention U-Net: Learning Where to Look for the Pancreas[J]. 2018.

[8] iek, zgün, Abdulkadir A , Lienkamp S S , et al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation[J]. Springer, Cham, 2016.