

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 课程报告



## 基于深度学习的手术器械分割

518021910971 裴奕博

## 目录

<b>1</b>	<b>项目简介</b>	<b>2</b>
1.1	项目背景 . . . . .	2
1.2	项目流程 . . . . .	2
1.3	项目任务 . . . . .	2
1.4	所用数据集 . . . . .	2
1.5	文件结构与功能 . . . . .	2
<b>2</b>	<b>实现方法</b>	<b>3</b>
2.1	数据预处理 . . . . .	3
2.1.1	图像的下采样 . . . . .	3
2.1.2	标签的分类 . . . . .	3
2.1.3	数据增强 . . . . .	3
2.2	网络结构实现 . . . . .	4
2.2.1	基于 UNet 的方法 . . . . .	4
2.2.2	基于 TDSNet 的多任务网络 . . . . .	5
2.3	训练参数的选取 . . . . .	6
<b>3</b>	<b>结果评估</b>	<b>6</b>
3.1	评价指标的选取 . . . . .	6
3.2	各方法评价指标对比 . . . . .	6
3.3	可视化结果对比 . . . . .	8
3.4	结果分析 . . . . .	8
<b>4</b>	<b>项目总结与展望</b>	<b>9</b>

## 1 项目简介

### 1.1 项目背景

如今，医疗机器人已经在外科手术中有了广泛的应用，并且正朝着更精细化和自主化的方向发展。在机器人辅助手术的一个关键部分涉及到手术器械的跟踪和姿态的确定。由于手术过程中，图像受到光照、阴影、反射的影响，从图像中分割出手术器械是一大挑战。基于深度学习的方法在这样的图像分割中已经取得了不错的效果，因此本项目通过深度学习的方法，试图分割手术过程图像中的手术器械。

### 1.2 项目流程

本项目的流程如下：

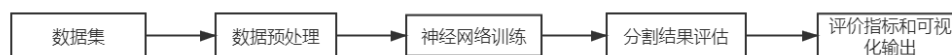


图 1: 项目工作流程图

整个流程和一般深度学习项目的流程相同，比较简单。

### 1.3 项目任务

本项目通过训练神经网络，实现对“binary”、“parts”和“instruments”三类问题在一定程度上的解决。其中，“binary”需要输出一张二值图像，仅代表器械与非器械。“parts”则需要表明其中一种器械主要器械的各个部分。“instruments”类则将对不同种类的手术器械进行分割。

### 1.4 所用数据集

选用了 MICCAI endovis2017 数据集，包括从 instrument\_dataset\_1 至 7 的共 1800 个样本。原图像为 24 位色深的 1920\*1080 的彩色 png 图像，ground truth 为 256 个灰度级的 1920\*1080 的黑白图像，不同的灰度代表不同的器械类别。

### 1.5 文件结构与功能

- raw\_data 文件夹：存放手术器械图像和 ground truth 的原始数据。
- cropped\_train 文件夹：存放经过裁剪、重采样和灰度化预处理后的数据。
- runs 文件夹：存放模型训练过程中产生的所有数据，包括模型结构、参数和训练记录 log。
- prepare\_data.py 文件：实现数据的预处理。

- prepare\_train\_val.py 文件：规定用于训练和评估的数据集。
- dataset.py 文件：定义 Pytorch dataset 和相关的图像和标签读取函数。
- loss.py 文件：定义损失函数。
- validation.py 文件：实现模型的评价指标的计算。
- train.py 文件：模型训练的主程序。
- utils.py 文件：定义其他所需的子模块。
- train.sh：训练脚本。
- result.json：存放训练的原始结果。
- json\_to\_csv.py：处理产生的原始数据并将其转换为 csv 格式。
- result.csv：最终生成的评估结果表格。

## 2 实现方法

### 2.1 数据预处理

#### 2.1.1 图像的下采样

由于原图和 ground truth 均为  $1920 \times 1080$ ，单张图像像素点很多，如果直接输入神经网络会导致参数量过多且运行速度缓慢，因此在数据预处理时通过下采样将输入数据转化为  $640 \times 512$

#### 2.1.2 标签的分类

根据用户所要解决的问题，需要将标签进行对应的分类，可以看到在 cropped\_train 文件夹下分别对应三种问题类型的 mask。在数据预处理中即将其分开。对原数据集中的 Left ground truth 进行二值化后即可得到 binary masks，将 Left ground truth 重新映射到 (0,255) 即可得到 parts masks，而对其他文件夹的文件进行总和即可得到 instruments masks。

#### 2.1.3 数据增强

本次数据集样本量较少，为使得神经网络训练达到更好的效果，需要进行数据增强。本次项目采用了 padding，随机裁切，水平垂直翻转，正则化等数据增强方法，有效扩充了数据量。

## 2.2 网络结构实现

### 2.2.1 基于 UNet 的方法

根据 [1] 中提供的方法，复现了论文中的 UNet，UNet-11，UNet-16 和 LinkNet-34 四种方法。这四个网络均为基于 UNet 的全卷积网络。

- UNet 是一种经典的网络，前半部分通过 CNN 编码器提取特征，逐步下采样，后半部分通过反卷积的方式进行上采样，并增加了跳跃的连接以减慢梯度的下降，使网络能够更深。最后输出与原图大小一致的预测值，可以直接与 label 进行比较计算 loss，是图像分割中常见的基本网络。
- UNet-11 和 UNet-16 是对 UNet 的改进，两者均采用了 VGGNet 作为 encoder，UNet11 和 UNet15 也是根据两者使用的 encoder 为 VGG-11 和 VGG-16 命名。

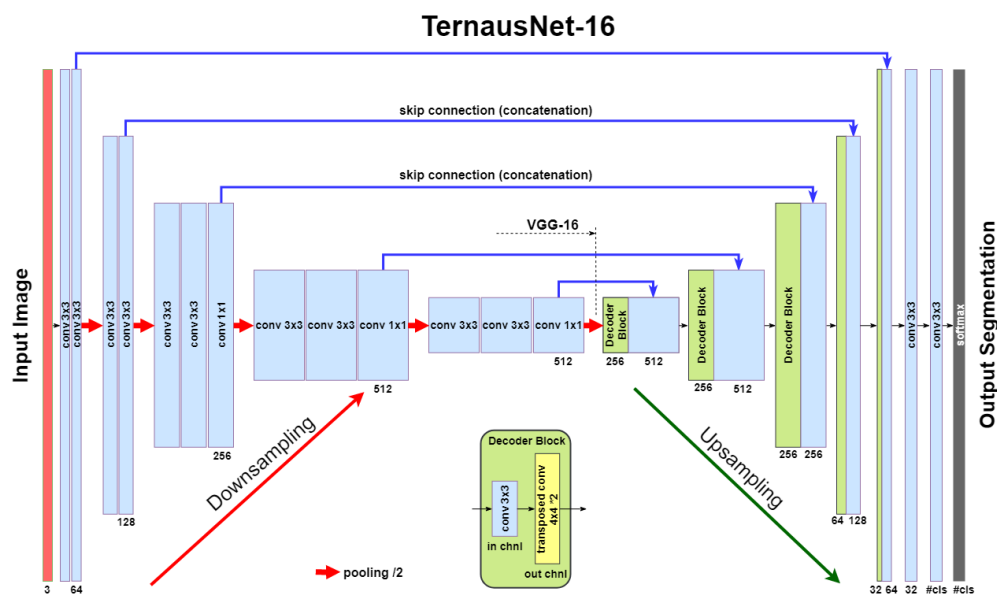


图 2: TernaUSNet 结构图

- LinkNet-34 是另一种对 UNet 的改进方法，采用了 ResNet-34 作为 encoder。此外在 decoder block 上也有了改进：添加了 batch normalization 和 1\*1 的卷积。

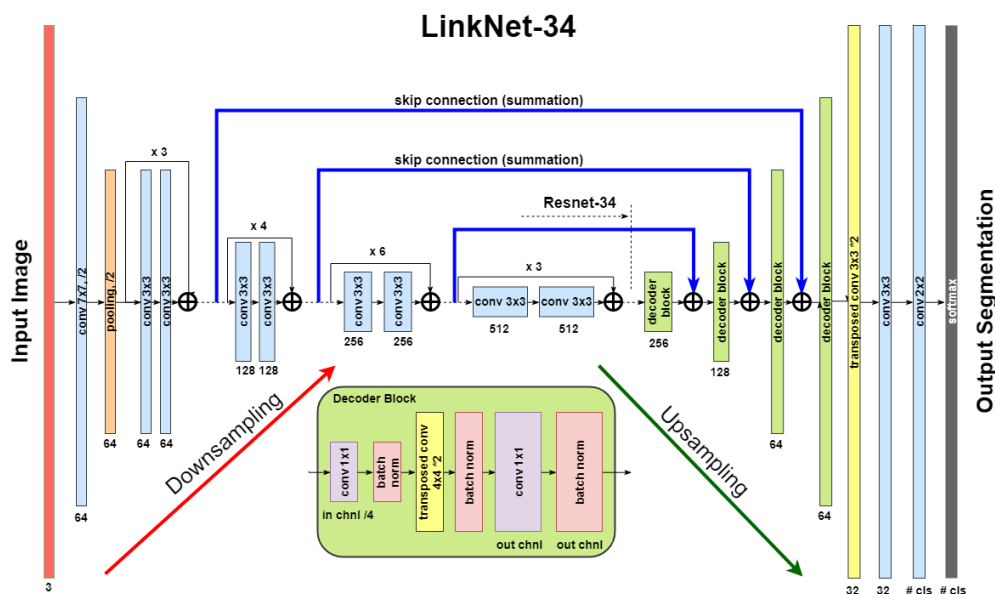


图 3: LinkNet 结构图

## 2.2.2 基于 TDSNet 的多任务网络

此外，由于以上几种网络对 instrument 类问题的分割效果并不好，因此考虑到各类问题之间的相互关联关系，设计了基于 TDSNet[2] 的多任务网络。

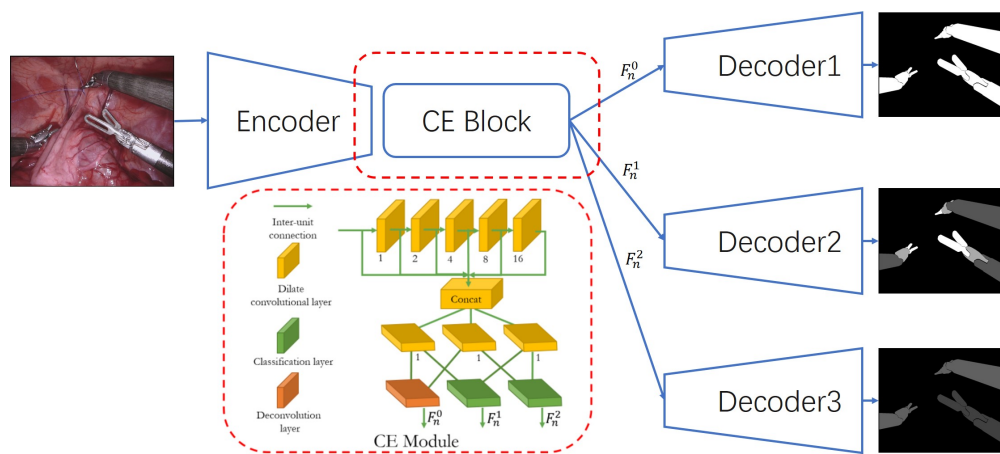


图 4: TDSNet 结构图

TDSNet 的要点在于在 encoder 和 decoder 之间插入了一个 CE Block，在这个 CE Block 中依次使用了不同步长的 dilate convolution 以整合不同大小的特征。随后将 CE Block 的输出  $F_n^0, F_n^1, F_n^2$  输入三个 decoder，并对应三种不同的分类任务。这样将三个不同的任务共用一个 encoder，在理论上可

以避免模型参数过多，并且在重点进行其中一个任务的分割时，可以有效利用到其他两个任务 ground truth 中所包含的信息，同时也提升了模型的运行速度。

## 2.3 训练参数的选取

- 选用  $L = H - k \log J$  作为损失函数，其中  $H$  为交叉熵 (cross entropy)， $k$  为常数（实际训练时取 0.3）， $J$  为 dice score，可以表示为

$$J = \frac{1}{n} \sum_{i=1}^n \left( \frac{y_i \hat{y}_i}{y_i + \hat{y}_i - y_i \hat{y}_i} \right)$$

在使用 TDSNet 时，损失函数为三者的加权和，实际使用时 binary,parts,instruments 三者的权重分别为 0.4,0.3,0.3

- 选用的优化器为 Adam，batch size 取 16（TDSNet 由于参数量大，取 8）。
- 学习率采用阶梯式，即前 10 个 epoch 采用  $1e-4$  使其更快收敛，后十个 epoch 采用  $1e-5$ 。
- 将数据集分为 [1, 3],[2, 5],[4, 8] [6, 7] 四个 fold，每次取其中一组作为 valid set 其余作为训练集，进行交叉验证算法。

## 3 结果评估

### 3.1 评价指标的选取

由于模型的输出和 ground truth 均为分割后的图像，因此评价指标的选取就比较简单，采用了图像分割中常用的 iou 和 dice 作为评价指标。定义预测为阳性的区域为  $A$ ，ground truth 中阳性区域为  $B$ ，则两者的计算公式如下：

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|} \quad (1)$$

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \quad (2)$$

此外，也同时输出 valid loss 作为对照

### 3.2 各方法评价指标对比

输出所用网络针对于各类问题的评价指标数据（TDSNet 的指标为单次运行时分出的数据）。输出表格如下：

表 1: 各模型评价指标对比

model	problem-type	valid_loss	iou	dice
LinkNet34	binary	0.291619	0.83452	0.90916
	instruments	39.55267	0.169484	0.254484
	parts	1.482577	0.752365	0.849873
TDSNet	binary	15.81483	0.440223	0.609769
	parts	15.81483	0.436848	0.590687
	instruments	15.81483	0.094368	0.159233
UNet	binary	0.372041	0.733065	0.845729
	instruments	41.29976	0.113255	0.18736
	parts	2.061818	0.57118	0.70922
UNet11	binary	0.264989	0.855534	0.921626
	instruments	40.18801	0.163311	0.252311
	parts	1.347551	0.739667	0.841694
UNet16	binary	0.252788	0.864523	0.927097
	instruments	40.19258	0.189746	0.277829
	parts	1.229443	0.776812	0.869413

可以看到以下几点:

- 最基本的 UNet 在 binary 问题上有不错的表现,但在其他问题上与改进后的其他网络有明显差距。
- 总体来说,对于 binary 和 parts 的分割都取得了不错的效果。UNet11, UNet16 和 LinkNet34 均达到了较高的 iou 和 dice。其中 UNet16 的效果最好,但网络结构也最为复杂,参数量最大,LinkNet 则在参数量减少的同时达到了不错的效果。
- TDSNet 表现不佳,在三个问题上均与其他网络有很大的差距。



### 3.3 可视化结果对比



图 5: 可视化结果对比图

- 可以看到，总体结果与 iou 和 dice 显示的指标一致，UNet11 和 UNet16 的表现是最好的。
- 还可以注意到，在 instruments 类的分割中，TDSNet 其实表现的没有评价指标显示的那么糟糕，基本分割出了边缘，评价指标低的原因是因为大面积的分错类，而这一现状在其他网络中多多少少也存在。

### 3.4 结果分析

可以看到，在 binary 问题和 parts 问题上，许多方法都达到了不错的效果，但对于 instruments 则效果不佳。产生这一结果的原因可能是数据集中各类样本的不平衡。查看数据集就会发现，部分手术器械的类别在数据集中很少出现，甚至在某些 fold 中整类器械一次都没有出现过，这对神经网络进行判断产生了很大的困难。与之对应的时，在最终输出的结果中，出现了大面积的分错类的现象，而非传统中更容易出现的分割边缘上的错误。这也解释了为什么神经网络在 parts 问题上表现良好而在 instruments 问题上发挥不佳。

TDSNet 在各类问题上表现不佳。推测原因可能有以下几点：

- 首先是模型的参数过多。在 Shvets 等人 [1] 的工作中，需要解决的问题是一个图像分割问题和两个 one-hot 的分类问题，而在本项目中，需要输出的是三个分割好的图像，在网络的后半部分需要三个结构相同的 decoder，因此相比原文中网络结构更为复杂，更难训练。

- 其次是在当前版本的 TDSNet 中, 我将 binary, parts 和 instruments 放在了同等重要的地位, 而事实上, binary 是其余两者的基础, parts 和 instruments 可以看作 binary 问题的进一步延伸。可能这样的差异会使网络的效果下降。
- 最后可能是由于时间有限, 模型的训练过程和各个超参数都未能调整到更佳, 在不改变网络结构的情况下通过改变超参数也可能会达到更好的效果。

## 4 项目总结与展望

本次项目通过深度学习的方法完成了对手术器械的分割, 在 binary 和 parts 任务上表现良好, 但对于 instruments 任务表现欠佳, 自己搭建的多任务网络也没有取得预期的效果, 只能说是部分完成了预期目标, 还有很多可以改进的空间。

尽管效果不尽如人意, 但是在本次大作业中我还是收获很多。本次大作业是我第一次真正有机会独立完成有关深度学习的项目。在完成项目的过程中, 我学习了一些关于深度学习的基本知识, Pytorch 框架的使用和一个深度学习项目基本的组成部分, 也完成了从原始数据集到模型训练验证的整个流程。根据老师给的参考资料上的思路, 也自己搭建了一个基于 TDSNet 的神经网络。虽然可能因为一些细节的问题导致效果不佳, 但是在完成项目的过程中我还是学到了很多, 并且接触到了实际的科研问题, 也算是对深度学习这个热门领域有了入门的了解。

最后模型实现的结果没能达到预期还是比较遗憾, 但是个人在完成项目的过程中也收获了很多。最后感谢老师和助教在本次项目完成过程中给予我的帮助!

## 参考文献

- [1] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, "Automatic instrument segmentation in robot-assisted surgery using deep learning,"
- [2] X. Ren, L. Zhang, S. Ahmad, D. Nie, F. Yang, L. Xiang, Q. Wang, and D. Shen, "Task decomposition and synchronization for semantic biomedical image segmentation," *arXiv preprint arXiv:1905.08720*, 2019.