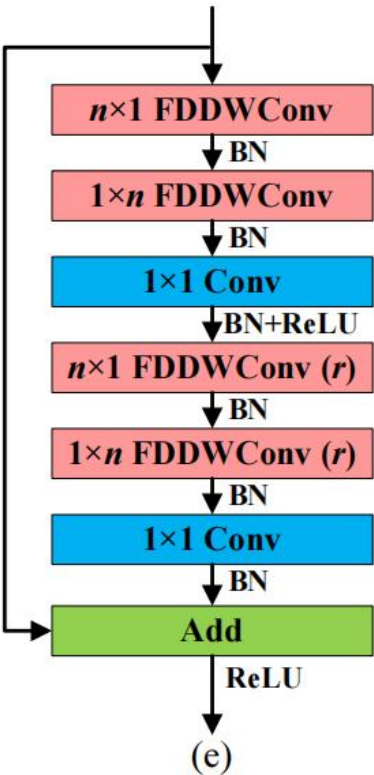


算法描述文档

模型架构

本次比赛中，我们使用了一种轻量级的语义分割和图像分割模型 FDDWNET，通过设计更深层的网络架构，同时保持更快的推理速度、更高的分割精度以及更小的模型参数。该网络使用分解扩张深度可分离卷积（FDDWC）来以较少的模型参数学习来自不同尺度感受野的特征表示。此外，FDDWNet 具有多个跳过连接的分支，以从中间卷积层收集上下文线索。实验表明，FDDWNet 仅具有 0.8M 的模型大小，但在分割精度和效果上较为出色。

FDDWNet 的核心单元是使用分解扩张深度可分离卷积（FDDWC）的极高效残差模块（extreme efficient Residual Module，EERM），它允许我们在少量模型参数的情况下从不同尺度的接受域学习特征表示。



EERM 模块架构图

Stage	Layer	Type	Size
Encoder	1	Downsampling Unit	$512 \times 256 \times 16$
	2	Downsampling Unit	$256 \times 128 \times 64$
	3-7	$5 \times \text{EERM} (r = 1)$	$256 \times 128 \times 64$
	8	Downsampling Unit	$128 \times 64 \times 128$
	9-16	$\begin{bmatrix} \text{EERM} (r = 1) \\ \text{EERM} (r = 2) \\ \text{EERM} (r = 5) \\ \text{EERM} (r = 9) \end{bmatrix} \times 2$	$128 \times 64 \times 128$
Decoder	17-24	$\begin{bmatrix} \text{EERM} (r = 2) \\ \text{EERM} (r = 5) \\ \text{EERM} (r = 9) \\ \text{EERM} (r = 17) \end{bmatrix} \times 2$	$128 \times 64 \times 128$
	25	Upsampling Unit	$256 \times 128 \times 64$
	26-27	$2 \times \text{EERM} (r = 1)$	$256 \times 128 \times 64$
	28	Upsampling Unit	$512 \times 256 \times 16$
	29-30	$2 \times \text{EERM} (r = 1)$	$512 \times 256 \times 16$
	31	Upsampling Unit	$1024 \times 512 \times C$

FDDWNet 架构图

模型架构的代码已在 model.py 文件中。

训练技术和学习策略

优化器部分，我们原先使用 Adam 优化器，后来使用了改进版的 **RAdam** 优化器，对初始学习率是具有鲁棒性的，可以适应更宽范围内的变化，拥有更好的训练效果。

在需要注意的**创新点**中，我们在训练过程中引入了**温度缩放**的策略，使用组合型的 **EnhancedCombinedLoss** 的损失函数,学习策略用的 **ReduceLROnPlateau** 进行动态学习率调整。

Configuration(我们的主要策略配置)

OPTIMIZER = 'RAdam' # 更改优化器名称

SCHEDULER = 'ReduceLROnPlateau'

SCHEDULER_PATIENCE = 2

SCHEDULER_FACTOR = 0.5

INITIAL_TEMPERATURE = 1.5

TEMP_LR_FACTOR = 0.1

epoch=200, batch_size=25

训练数据增强部分:

```
def get_train_transforms(self):
```

```
    return A.Compose([
```

```
        A.HorizontalFlip(p=0.5),
```

```
        A.VerticalFlip(p=0.5),
```

```
        A.RandomRotate90(p=0.5),
```

```
        A.RandomBrightnessContrast(p=0.5),
```

```
        A.Normalize(
```

```
            mean=[0.485, 0.456, 0.406],
```

```
            std=[0.229, 0.224, 0.225],
```

```
            max_pixel_value=255.0
```

```
        ),
```

```
        ToTensorV2(), ])
```

温度缩放策略（Temperature Scaling）：

温度标度是一种校正神经网络的后处理技术。经过温度缩放后，可以使神经网络输出的概率更加平滑和可信。它作为一个轻量级模块，用于辅助训练，不作为实际网络的一部分确实可以较好提升模型实际效果。

简单例子：

实际例子

假设其中一个 logits 值在某个像素位置 (x, y) 为：

- 类别1: 2.0
- 类别2: 0.5
- 类别3: -1.0
- 类别4: 1.5

经过 AdaptiveTemperatureScaling 模块调整后：

1. 计算出该位置的有效温度，比如类别1温度为1.2，类别2为0.8，类别3为1.5，类别4为1.0。
2. 对每个类别的 logits 进行缩放：

- 类别1的 $\text{scaled_logit} = 2.0 / 1.2 \approx 1.67$
- 类别2的 $\text{scaled_logit} = 0.5 / 0.8 \approx 0.625$
- 类别3的 $\text{scaled_logit} = -1.0 / 1.5 \approx -0.67$
- 类别4的 $\text{scaled_logit} = 1.5 / 1.0 = 1.5$

最终，这些调整过的 logits 将用于后续的 softmax 计算，提升模型在置信度估计和分类性能上的表现。

值得注意的是，我们在该策略中引入了**注意力机制**，代码中 `spatial_attention`，空间注意力机制，通过卷积层和激活函数生成空间注意力权重，帮助模型关注输入图像中的重要区域。`channel_attention`，通道注意力模块，使用全局平均池化和卷积层生成通道注意力权重，帮助模型关注重要的特征通道。我们定义的 `AdaptiveTemperatureScaling` 这个模块通过结合空间注意力、通道注意力和置信度权重，动态地调整每个类别的温度参数，从而提高了模型的校准能力。

在本次比赛中，我们首先获取每个类别的温度值用于监控，之后应用温度调节输出，梯度裁剪，防止温度值剧烈波动，同时也对温度模块使用较小的学习率。防止模型了太过自信，平滑了模型的分割概率。具体代码实现可以看我们在后面放出的网盘链接文件。

EnhancedCombinedLoss 损失函数：

这种**组合性**损失函数的设计旨在提高模型在图像分割任务中的性能，特别是通过关注边界区域来提高分割精度。通过调整 `alpha` 和 `beta` 参数，可以平衡不同损失项的贡献，以适应不同的数据集和任务需求。

loss = 0.5(focal_loss+dice_loss)+0.5 boundary_loss

我们定义了四个类，分别为 DiceLoss、FocalLoss、CombinedLoss 和 EnhancedCombinedLoss，最后使用的 EnhancedCombinedLoss 结合了 DICEloss、focalloss 和 boundaryloss.。Dice 损失主要关注整体的分割质量，适用于衡量整体相似性。在 DiceLoss 中，我们使用了标签平滑（Label Smoothing）的小技巧这在一定程度可以抑制网络过拟合的现象。Focal 损失则针对类别不平衡问题，通过降低易分类样本的权重，增强对难分类样本的关注。边界损失特别强调边界区域的准确性，确保分割结果在边界处更加精细。那个边界损失可以理解成 BCEloss 的进阶版。

EnhancedCombinedLoss 的_init_()参数:

alpha: Dice#损失和 Focal 损失的权重比，用于平衡两者在总损失中的贡献。

beta: #边界损失的权重，用于调整边界损失对总损失的影响。

dice_loss: Dice#损失实例，用于计算预测和真实标签之间的 Dice 系数损失。

focal_loss: Focal#损失实例，用于处理类别不平衡问题，特别关注难分类的样本。






总损失由 Dice 损失、Focal 损失和边界损失组成，通过 alpha 和 beta 参数进行加权。alpha 用于平衡 Dice 损失和 Focal 损失，而 beta 用于调整边界损失的影响。这种组合损失函数的设计旨在提高模型在图像分割任务中的性能，特别是通过关注边界区域来提高分割精度。通过调整 alpha 和 beta 参数，可以平衡不同损失项的贡献，以适应不同的数据集和任务需求。

具体实现请看我们在放出的网盘链接文件。

我们的网盘代码链接:

https://pan.baidu.com/s/1ZV86k5xminGwMQ2cXPLK_Q

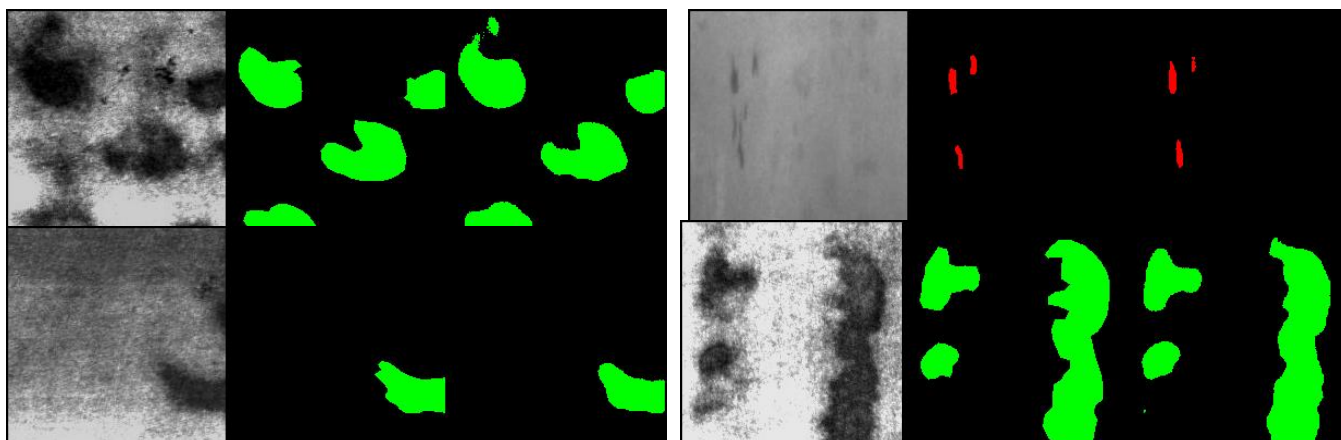
提取码: 1111

 data.py	2024/10/25 19:40	Python 源文件	3 KB
 loss.py	2024/10/25 10:43	Python 源文件	6 KB
 Net_model.pth	2024/10/25 19:31	PTH 文件	3,658 KB
 radam.py	2024/10/25 10:43	Python 源文件	11 KB
 temperature.py	2024/10/25 10:43	Python 源文件	5 KB
 train.py	2024/10/25 10:50	Python 源文件	7 KB

data.py, 数据读取代码 loss.py EnhancedCombinedLoss 损失函数代码

radam.py 优化器代码 temperature.py 温度缩放代码 train.py 训练代码

部分划分效果图片



参考文献

模型架构:

FDDWNET: A LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORK FOR REAL-TIME SEMANTIC SEGMENTATION

Jia Liu¹, Quan Zhou^{1,*}, Yong Qiang¹, Bin Kang², Xiaofu Wu¹, and Baoyu Zheng¹

National Engineering Research Center of Communications and Networking,

Nanjing University of Posts & Telecommunications, P.R. China.

School of Internet of Things, Nanjing University of Posts & Telecommunications, P.R. China.

ABSTRACT

This paper introduces a lightweight convolutional neural network, called *FDDWNet*, for real-time accurate semantic segmentation. In contrast to recent advances of lightweight networks that prefer to utilize shallow structure, *FDDWNet* makes an effort to design more deeper network architecture, while maintains faster inference speed and higher segmentation accuracy. Our network uses factorized dilated depth-wise separable convolutions (FDDWC) to learn feature representations from different scale receptive fields with fewer model parameters. Additionally, *FDDWNet* has multiple branches of skipped connections to gather context cues from intermediate convolution layers. The experiments show that *FDDWNet* only has 0.8M model size, while achieves 60 FPS running speed on a single RTX 2080Ti GPU with a 1024×512 input image. The comprehensive experiments demonstrate that our model achieves state-of-the-art results in terms of available speed and accuracy trade-off on CityScapes and CamVid datasets.

Index Terms— Lightweight network, Semantic segmentation, Factorized convolution, Dilated depthwise convolution

摘要

本文介绍了一种用于实时准确语义分割的轻量级卷积神经网络 FDDWNet。相对于近来轻量网络倾向于使用浅层结构，FDDWNet 致力于设计更深层次的网络架构，同时保持更快的推理速度和更高的分割精度。我们的网络使用分解扩张深度可分离卷积（FDDWC）来学习来自不同规模接受域的特征表示，模型参数更少。此外，FDDWNet 有多个跳过连接的分支，用于从中间卷积层收集上下文线索。实验表明，FDDWNet 模型大小仅为 0.8M，而在单个 RTX 2080Ti GPU 上，以 1024×512 输入图像实现 60 FPS 的运行速度。综合实验表明，我们的模型在 cityscapes 和 CamVid 数据集上的可用速度和精度权衡方面达到了最先进的结果。

索引术语：轻量级网络，语义段卷积，分解卷积，扩展深度卷积

温度缩放: https://github.com/gpleiss/temperature_scaling

<https://arxiv.org/abs/1706.04599>

标签平滑: <https://github.com/ankandrew/online-label-smoothing-pt>

<https://arxiv.org/abs/2011.12562>