论文复现：Rethinking BiSeNet For Real-time Semantic Segmentation

摘要

BiSeNet 已被证明是一种流行的用于实时分割的双流网络。它主要由两条路径构成：一个上下文路径来编码不同感受野和不同尺度的高级语义信息，一个空间路径来编码丰富的细节空间信息。最后使用特征融合模块把两个信息融合得到最后的准确的预测结果。然而,它的增加额外路径编码空间信息的原理是费时的，并且额外路径总是缺乏低层次的信息指导，无法有效分割图像，故提出 Rethinking BiSeNet。针对 BiSeNet，作者主要进行了两个方面的改进：针对 BiSeNet 中的上下文路径 ，作者改进了它编码部分，针对空间路径，作者细节指导来替代它。

关键词：BiSeNet，STDC 网络，上下文路径，空间路径

1 引言

语义分割是计算机视觉领域的一个经典和基础的问题，旨在为图像分配像素级的类别标签。深度学习的快速发展极大地促进了语义分割的性能，图像的语义分割在自动驾驶、视频监控、机器人传感等许多应用领域都有着重要的应用。由于本人的研究方向是图像分割，所以选择对这篇图像分割的论文进行复现。这不但可以帮助我进一步加深对于自己研究方向的理解，而且锻炼自己的动手实践的能力。同时选择的论文是最近几年的，可以很好的学习一些前沿的知识。

2 相关工作

2.1 BiSeNet 的提出

在语义分割领域，由于需要对输入图片进行逐像素的分类，运算量很大。之前的实时性语义分割算法，主要有三种加速方法：

（1）通过剪裁或 resize 来限定输入大小，以降低计算复杂度(图 1(a)的第一个图)。优点：简单而有效。缺点：空间细节丢失，尤其是边界部分，导致度量和可视化的精度下降。

（2）通过减少网络通道数量加快处理速度，即使用轻量级网络，缺点：弱化空间信息（图 1(a)的第二个图）。

（3）为追求极其紧凑的框架而丢弃模型的最后阶段的下采样，缺点：由于抛弃了最后阶段的下采样，模型的感受野不足以涵盖大物体，导致判别能力较差。为了解决上述的问题，提出了 BiSeNet（图 1(c)）。

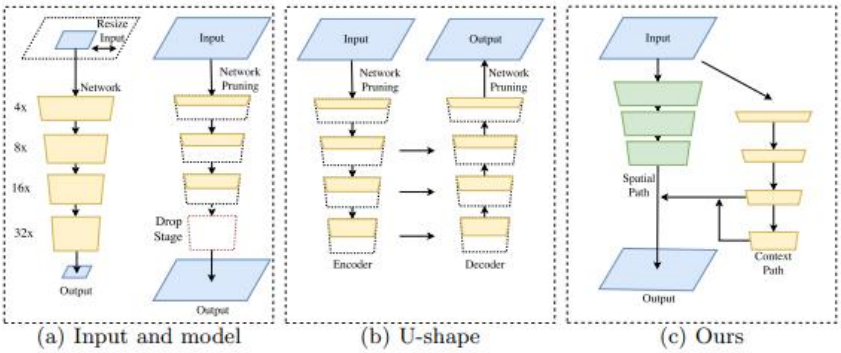

图 1：不同网络的模型

2.2 BiSeNet 网络结构

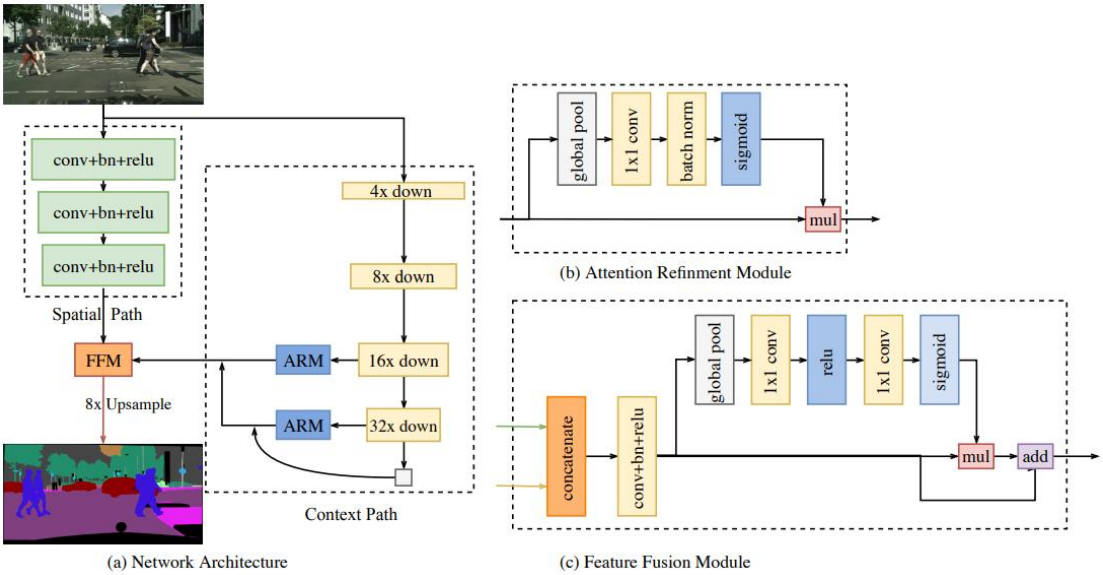

图 2：BiSeNet 网络结构图

它由两条路径组成，一个 Context path 编码不同感受野和不同尺度的高级语义信息（即 high-level feature），一个 Spatial path 编码丰富的细节空间信息（即 low-level feature），最后使用 FFM（特征融合模块）把两个信息融合得到最后的预测结果。但是由于 BiSeNet 中添加额外 path 以对空间信息进行编码很耗时，并且额外 path 总是缺乏低层次的信息指导，无法有效分割图像，故提出 Rethinking BiSeNet。

3 本文方法

3.1 本文方法概述

BiSeNet 添加一条额外的路径来获取低层次特征是很费时的，同时这条路径也往往缺乏低层次特征信息的引导，作者改进了这个耗时的空间路径，如下图：

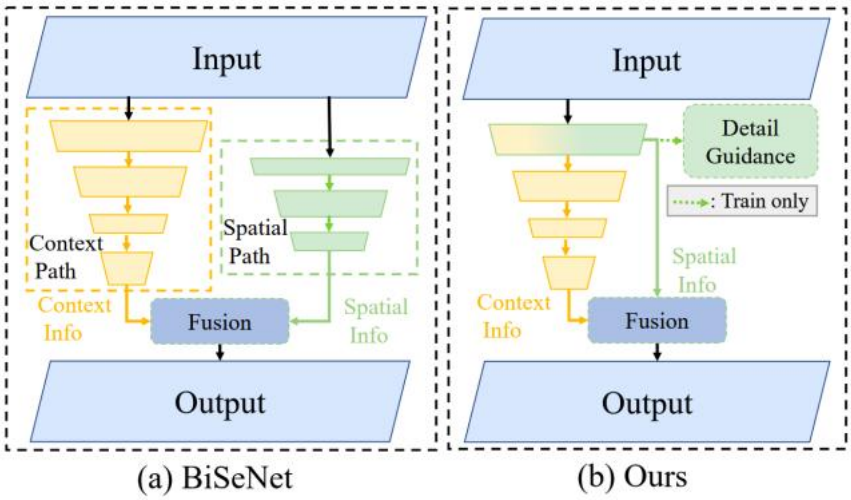

图 3：BiSeNet 与改进的 BiSeNet

针对 BiSeNet 中的 Context path ，作者改进了它的编码部分，针对 BiSeNet 中的 Spatial path，作者提出了 Detail guidance 来替代它。

STDC Segmentation 网络结构如下图 4 所示。

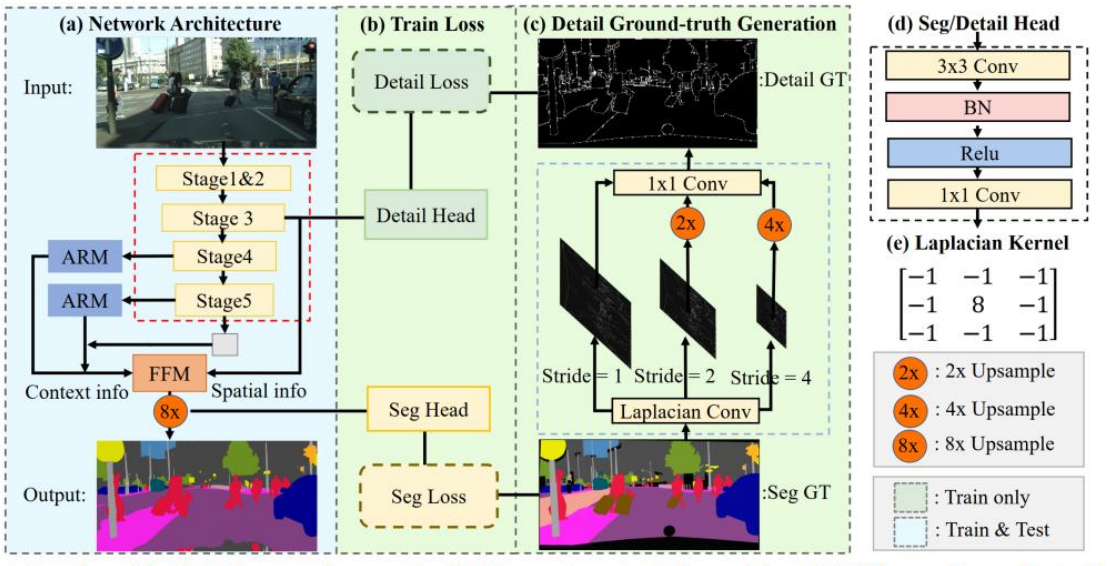

图 4：STDC Segmentation 网络结构

ARM 表示注意提取模块，FFM 表示特征融合模块。红色虚线框中的操作是 STDC 网络。蓝色虚线框中的操作是细节聚合模块。

3.2 改进 Context path 中的编码部分

这部分就是针对 BiSeNet 中的 Context path 中的网络进行改进，作者设计了自己的 STDC 网络，网络由 STDC 模块组成。如下图：

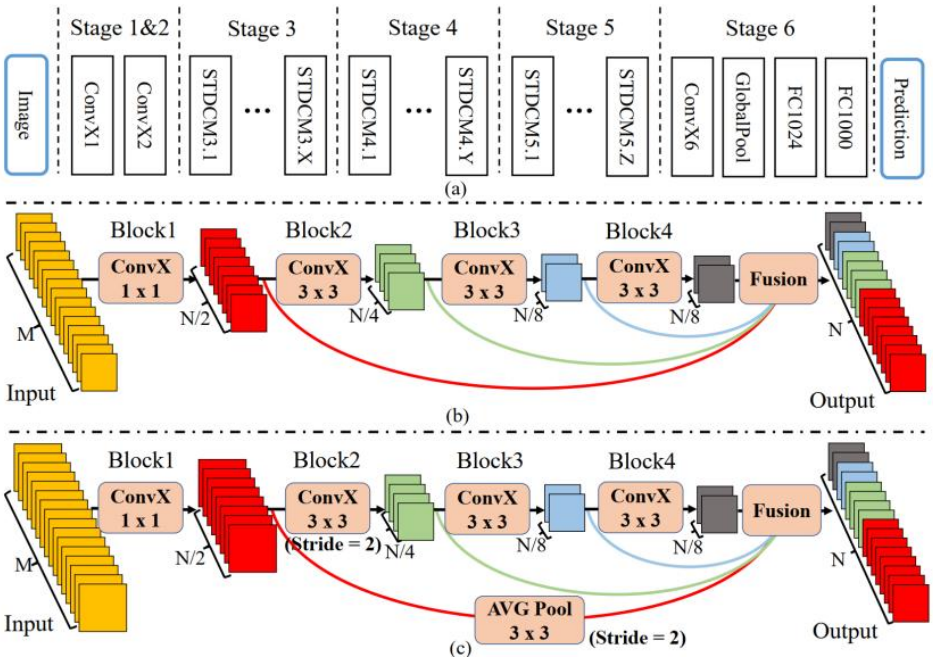

图 5：STDC 网络架构

(a)一般 STDC 网络架构。ConvX operation 是指 convn - bn - relu。(b)网络中使用的短期密集连接模块(STDC 模块)。M 表示输入通道维数，N 为输出维数频道。每个块都是一个具有不同内核的 ConvX 操作大小。(c) STDC 模块，stride=2。将 STDC 模块集成到 U-net 体系结构中，形成 STDC Network，提高了语义分割任务网络的性能。

3.3 用细节指导代替 Spatial path

这部分就是针对 BiSeNet 中的 Spatial path 的改进，作者设计了 Detail guidance 来替代 BiSeNet 中的 Spatial path。

3.3.1 Segmentation Architecture

使用预训练的 STDC 网络作为编码器的 Backbone，并采用 BiSeNet 的 Context path 对 Context 信息进行编码。

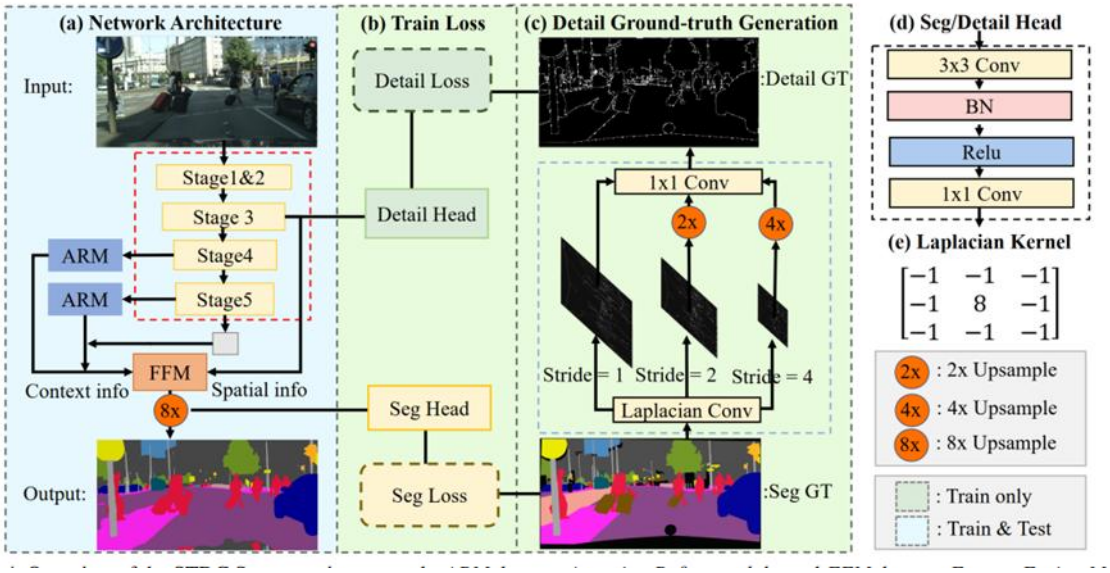

图 6：网络分割结构

如图 6(a)所示，作者使用 stage 3、4、5 来生成下采样比率分别为 1/8、1/16、1/32 的特征图。然后使用全局平均池化得到语义信息。然后，使用 U-shape 结构来对全局特征进行上采样，并且和 stage4、stage5 的进行结合（在 encoder 阶段）。在最后的语义分割预测阶段，作者使用了 特征融合模块 FFM，来融合来自 encoder 的 stage3 (1/8 大小) 和 decoder 的 stage3 的特征，作者认为来自这两个 stage 的特征代表了不同尺度的特征。encoding 的特征有更多的细节信息，decoding 的特征有更多的语义信息（由于其来自于 global average pooling）。

Seg Head 的构成：一个 3x3 conv+bn+relu，再跟一个 1x1 卷积，输出维度为类别数量，其中所使用的 loss 是交叉熵损失。

3.3.2 Detail Guidance of Low-level Features

BiSeNet 的 spatial path 的特征如下图所示，对比 STDC Network 的低层特征（如 stage3），spatial path 包含了更多细节信息，如边缘、角点。与图 7(c)相比，(d)由于添加了 Detail Guidance
所以包含了更多的细节信息。

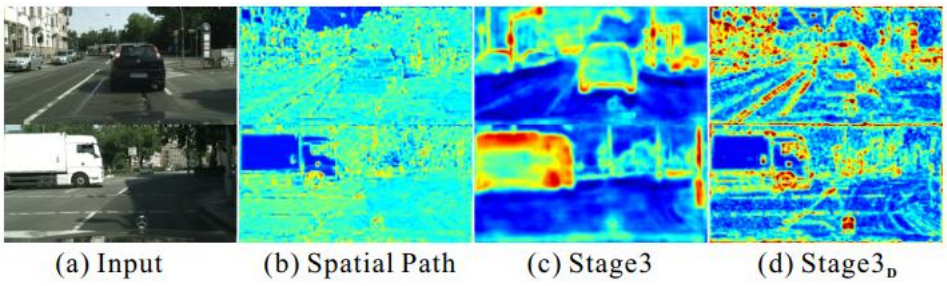

图 7：不同网络结构获取的细节信息

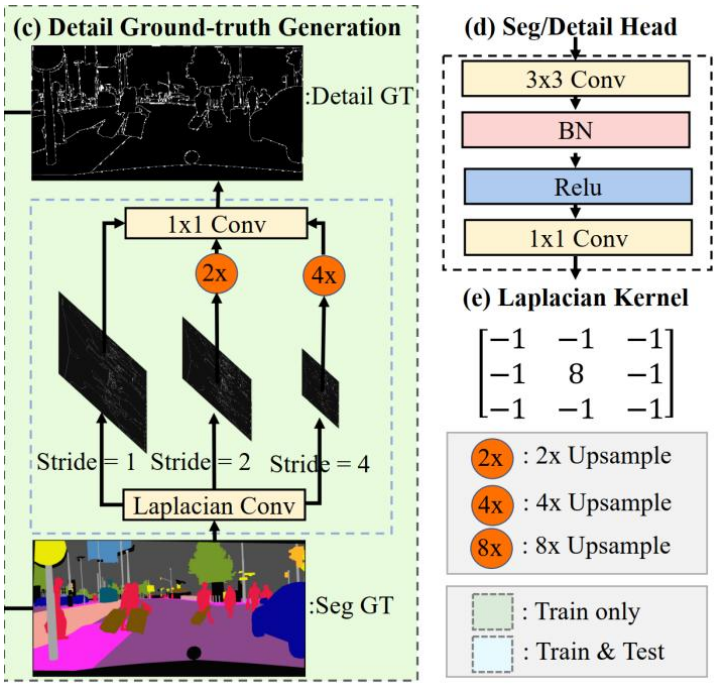

图 8：通过 detail Aggregation 模块从语义分割 ground truth 中生成 binary detail ground-truth

首先使用不同 stride 的 2D 卷积（laplacian kernel，图 8(e)），产生不同尺度的 soft thin detail 特征图。然后将这三个特征图上采样到原始尺寸，并使用一个 1x1 卷积进行动态融合。最后，使用阈值 0.1 来将预测结果转化为二值图。Detail Head 产生 detail map，detail map 可以指导浅层对空间信息编码

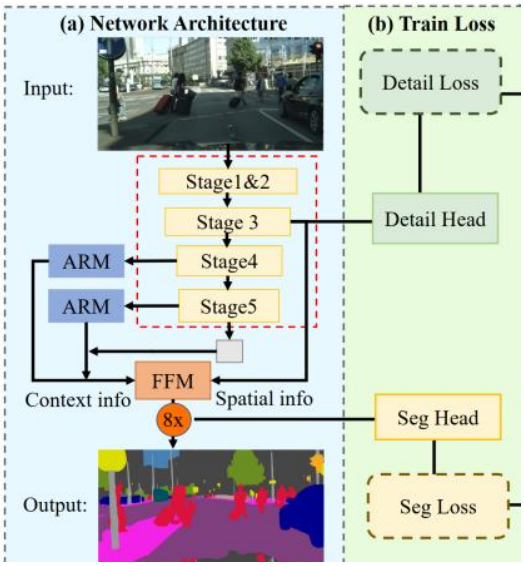

图 9：detail map 可以指导浅层对空间信息编码

与之前一样采用 U-shape 结构来对全局特征进行上采样，并且和 stage4、stage5 的进行结合得到 Context info（上下文信息），然后再和 Detail Head 产生 Spatial info（空间信息）进行融合，最后经过 8 倍上采样得到最终的分割结果。

3.4 损失函数定义

Detail 损失函数：

$$L_{detail}(p_d, g_d) = L_{dice}(p_d, g_d) + L_{bce}(p_d, g_d)$$

$$L_{dice}(p_d, g_d) = 1 - \frac{2\sum_i^{H\times W} p_d^i g_d^i + \epsilon}{\sum_i^{H\times W} (p_d^i)^2 + \sum_i^{H\times W} (g_d^i)^2 + \epsilon}$$

其中$p_d \in R^{H\times W}$为预测细节，$g_d \in R^{H\times W}$表示 ground-truth 对应的细节。L_{bce}为二元交叉熵损失，L_{dice}为 dice 损失。总的损失函数为最后预测值与真实值的损失函数加上L_{detail}

4 复现细节

4.1 与已有开源代码对比

1、引用代码：https://github.com/MichaelFan01/STDC-Seg，代码为官方提供的代码。

2、使用情况：使用了作者给的代码的所有模块。

3、自己的工作量：首先，修改了代码中 modules 模块下的 function 中的反向传播 backward 的返回参数，使代码能够正常运行。其次，论文中并没有将分

割的结果可视化，所以增加了一个可视化功能 predict 将图像分割的结果进行可视化，使得复现的效果更加直观。

4.2 使用环境搭建

- Pytorch 1.7.0
- Python 3.7
- NVIDIA GPU（V100）
- TensorRT 8.5.1.7

4.3 创新点

作者对 BiSeNet 的 Context path 和 Spatial path 进行了改进，使用 STDC 网络作为 Context path 的骨干网络。对于 Spatial path，作者提出了一个细节聚合(Detail Aggrega)模块，将空间信息的学习以单流的方式集成到底层。最后，将底层特征和深层特征进行融合，得到比之前更准确的分割结果。

5 实验结果分析

5.1 实施细节

1、采用 Cityscapes 数据集，分为训练集、验证集和测试集，图片数量分别为 2975，500，1225。

2、评价标准：平均交并比（mIoU），每秒传输帧数（Frames Per Second）)

3、Context path 中分别采用了 STDC1 和 STDC2 作为编码器进行图像的语义分割，参数如下表所示。

Stages	Output size	KSize	S	STDC1		STDC2	
				R	C	R	C
Image	224×224				3		3
ConvX1	112×112	3×3	2	1	32	1	32
ConvX2	56×56	3×3	2	1	64	1	64
Stage3	28×28 28×28		2 1	1 1	256	1 3	256
Stage4	14×14 14×14		2 1	1 1	512	1 4	512
Stage5	7×7 7×7		2 1	1 1	1024	1 2	1024
ConvX6	7×7	1×1	1	1	1024	1	1024
GlobalPool	1×1	7×7					
FC1					1024		1024
FC2					1000		1000
FLOPs				813M		1446M	
Params				8.44M		12.47M	

4、使用一块 V100 的 Gpu，训练次数是 120000。

5.2 模型评估

1、官方给的 STDCSeg 模型评估：

Model	Resolution	mIoU(%)	FPS
STDC1-Seg50	512×1024	0.722	156.2
STDC1-Seg75	768×1536	0.745	81.8
STDC2-Seg50	512×1024	0.742	113.3
STDC2-Seg75	768×1536	0.770	68.6

2、复现的 train_STDCSeg 模型评估：

Model	Resolution	mIoU(%)	FPS
train_STDC1-Seg50	512×1024	0.700	153.6
train_STDC1-Seg75	768×1536	0.739	80.3
train_STDC2-Seg50	512×1024	0.730	118.3
train_STDC2-Seg75	768×1536	0.761	67.3

其中 50 和 75 分别代表图片输入大小分别为 512*1024，768*1536。

5.3 可视化结果

从模型评估的结果来看，无论是官方给的模型，还是复现出来的模型，输入大小为 75 的模型的 mIOU 都高于输入大小为 50 的模型，故本文选择对输入大小为 75 的模型进行可视化，STDC1_75 表示官方训练的模型，train_STDC1_75 表示

本文复现的模型。

验证集可视化结果：

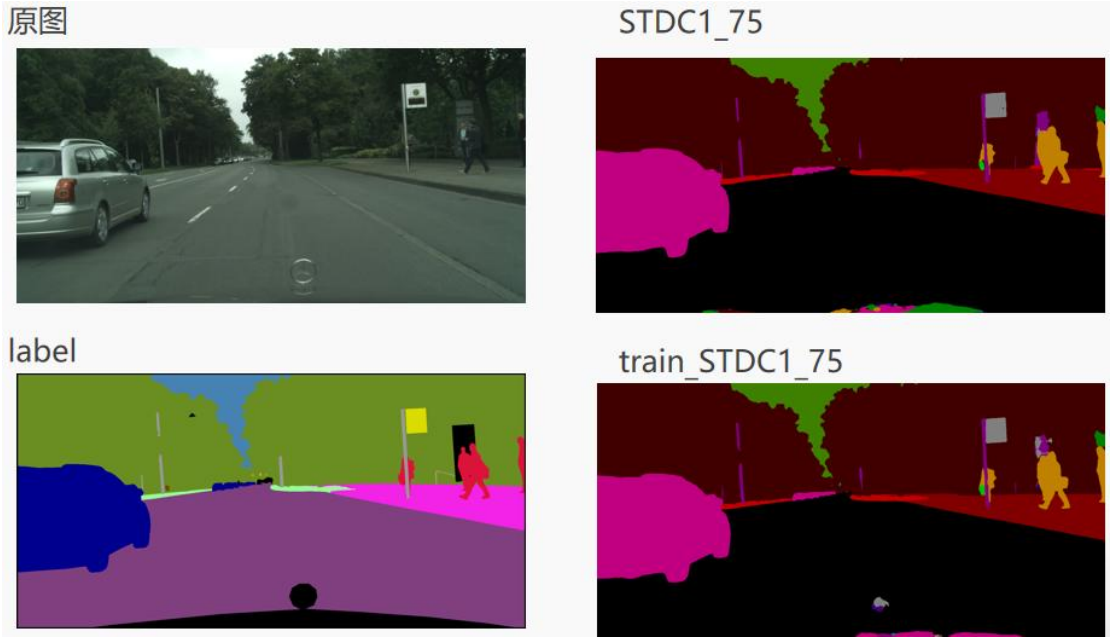

图 10：验证集 STDC1_75 与 train_STDC1_75 可视化结果

图 11：验证集 STDC2_75 与 train_STDC2_75 可视化结果

测试集可视化结果：

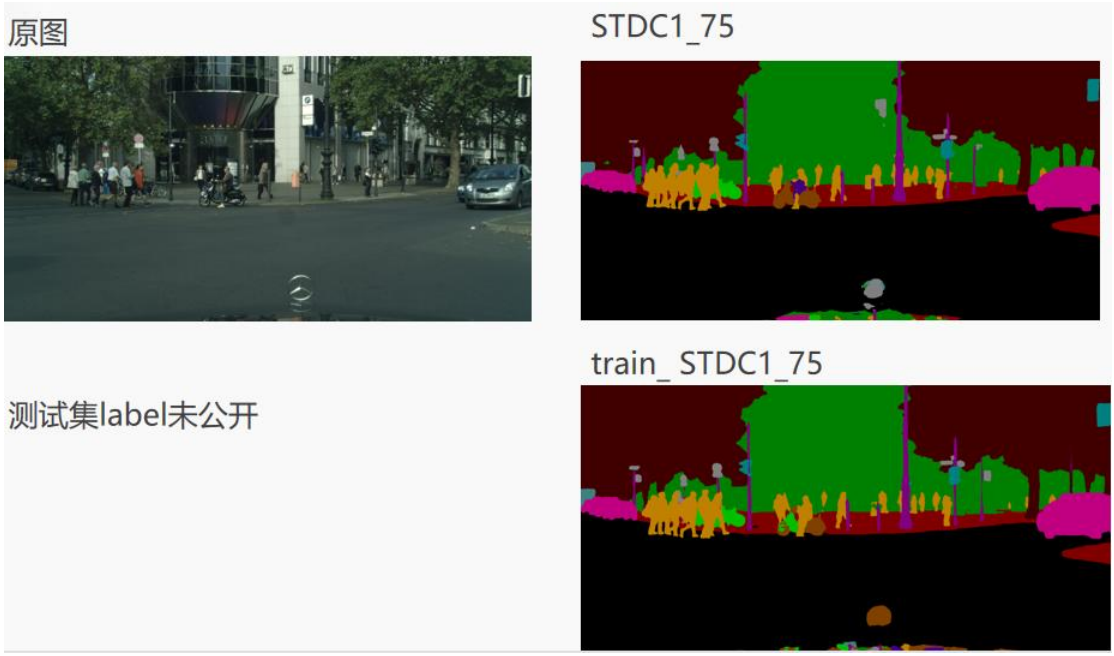

图 12：测试集 STDC1_75 与 train_STDC1_75 可视化结果

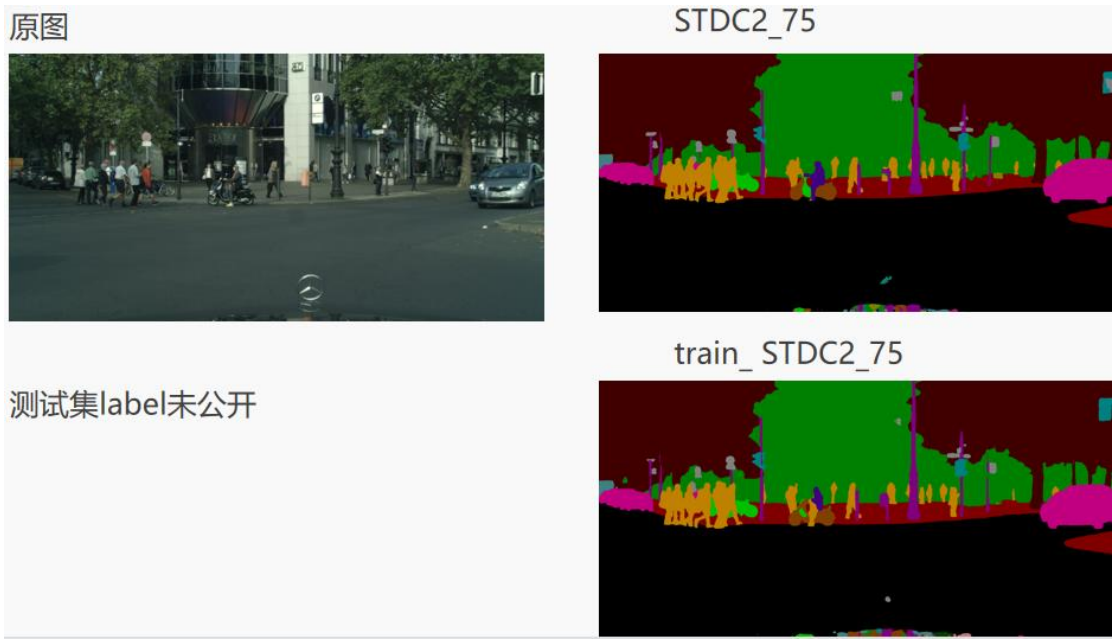

图 13：测试集 STDC2_75 与 train_STDC2_75 可视化结果

6 总结与展望

6.1 总结

本文首先介绍了改进的 BiSeNet 的相关工作，其次是所使用到的技术，然后是复现的细节，最后是复现结果的展示。由于 BiSeNet 的空间路径比较耗时，且网络并不能有效地进行图像分割。为了解决这些问题，该网络被命名为 STDC 网络。具体来说，作者逐步降低特征图的维数，利用特征图的聚合进行图像表示。在解码器中，作者提出了一个细节聚合模块，将空间信息的学习以单流的方式集成到底层。最后，将底层特征和深层特征进行融合，得到最终的分割结果。作者在 Cityscape 和 CamVid 数据集上的大量实验证明了该方法的有效性，在分割精度和推理速度之间实现了良好的平衡。

6.2 展望

本文复现的模型也有一些不足之处。首先，从我们复现的结果中可以看出，复现的模型的 mIOU 要比官方的模型低一点，可能训练的次数还不够，损失函数还没有完全收敛。其次是虽然整体的分割效果还不错，但是从可视化结果中也不难看出，每一幅图都有几处没有分割好，这是需要优化的地方。从之前的结果中可以看出，输入的图像大一点，模型的 mIOU 就会高一点，可以考虑增大输入尺寸，但是也要考虑计算量。也可以考虑对网络的结构进行调整，进一步优化网络结构。

参考文献

[1] Mingyuan F, Shenqi L, Junshi H, Xiaoming W, Zhenhua C, Junfeng L, Xiaolin W, et al. Rethinking BiSeNet For Real-time Semantic Segmentation[C], Computer Vision and Pattern Recognition, 2021: 9716-9725.