BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation

语义分割需要丰富的空间信息和相当大的感受野。然而，现代方法通常会牺牲空间分辨率来实现实时推理速度，这导致了性能不佳。在本文中，我们用一个新颖的双边分割网络（BiSeNet）来解决这个难题。我们首先设计了一个小步幅的空间路径，以保留空间信息并产生高分辨率的特征。同时，采用快速下采样策略的上下文路径，以获得足够的感受野。在这两条路径的基础上，我们引入了一个新的特征融合模块来有效地结合特征。在Cityscapes、CamVid和COCO-Stuff数据集上，提议的架构在速度和分割性能之间取得了适当的平衡。具体来说，对于2048×1024的输入，我们在Cityscapes测试数据集上实现了68.4%的平均IOU，在一块NVIDIA Titan XP卡上的速度为105FPS，这比现有的具有可比性的方法要快得多。

1. 介绍

语义分割的研究，相当于给每个像素分配语义标签，是计算机视觉中的一项基本任务。它可以广泛地应用于增强现实设备、自动驾驶和视频监控等领域。这些应用对高效的推理速度有很高的要求，以实现快速互动或响应。

最近，实时语义分割的算法[1,17,25,39]表明，主要有三种方法来加速模型。1）[34,39]试图通过裁剪或调整大小来限制输入尺寸以降低计算的复杂性。虽然该方法简单有效，但空间细节的损失破坏了预测，特别是在边界周围，导致指标和可视化的准确性下降。2) 有些工作没有调整输入图像的大小，而是修剪网络的通道以提高推理速度[1, 8, 25]，特别是在基础模型的早期阶段。然而，它削弱了空间容量。3) 对于最后一种情况，ENet[25]提出放弃模型的最后阶段，以追求一个极其紧密的框架。然而，这种方法的缺点是显而易见的：由于ENet放弃了最后阶段的下采样操作，模型的接受域不足以覆盖大型物体，导致辨别能力差。总的来说，上述所有的方法都向速度妥协，这在实践中是比较差的。图1（a）给出了说明。为了弥补上述空间细节的损失，研究人员广泛利用U型结构[1,25,35]。通过融合骨干网络的层次特征，U型结构逐渐提高了空间分辨率，填补了一些缺失的细节。然而，这种技术有两个弱点。1）完整的U型结构会降低模型的速度，因为引入了高分辨率特征图的额外计算。2）更重要的是，如图1（b）所示，在修剪或裁剪中丢失的大部分空间信息不能通过涉及浅层而轻易恢复。换句话说，U型技术最好被看作是一种救济，而不是一种基本的解决方案。基于上述观察，我们提出了双侧分割网络（BiSeNet），包括两部分。空间路径（SP）和背景路径（CP）。顾名思义，这两个部分分别是为了应对空间信息的损失和感受野的萎缩而设计的。这两个路径的设计理念是明确的。对于空间路径，我们只堆叠三个卷积层来获得1/8的特征图，它保留了丰富的空间细节。对于Context Path，我们在Xception[8]的尾部附加了一个全局平均池化层，其中的感受野是骨干网络的最大值。图1（c）显示了这两个部分的结构。为了追求更好的准确性而不损失速度，我们还研究了两条路径的融合和最终预测的细化，并分别提出了特征融合模块（FFM）和注意力细化模块（ARM）。正如我们下面的实验所示，这两个额外的组件可以进一步提高Cityscapes[9]、CamVid[2]和COCO-Stuff[3]两个基准的整体语义分割精度。我们的主要贡献总结如下。

- 我们提出了一种新颖的方法，将空间信息保存和接受场提供的功能解耦为两条路径。具体来说，我们提出了一个具有空间路径（SP）和上下文路径（CP）的双边分割网络（BiSeNet）。

- 我们设计了两个特定的模块，特征融合模块（FFM）和注意力细化模块（ARM），以可接受的成本进一步提高准确性。

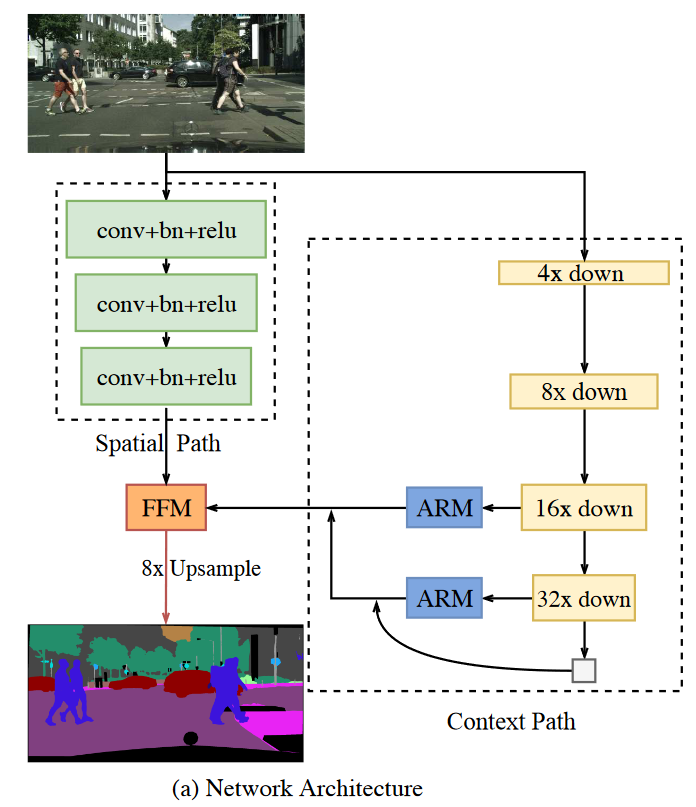
- 我们在Cityscapes、CamVid和COCO-Stuff等基准测试中取得了令人印象深刻的结果。更具体地说，我们在Cityscapes测试数据集上取得了68.4%的结果，速度为105FPS。

3.网络

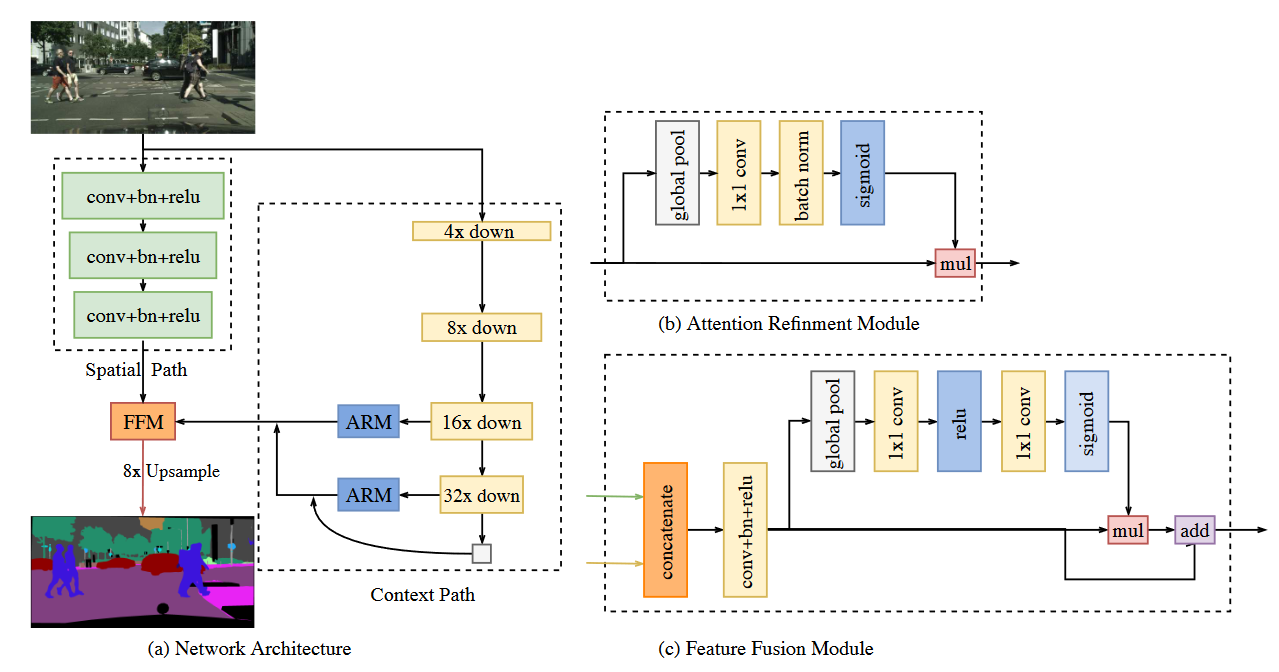
在这一节中，我们首先详细说明了我们提出的双边分割网络（BiSeNet）的空间路径和背景路径。此外，我们还详细说明了这两条路径的相应有效性。最后，我们展示了如何将这两条路径的特征与特征融合模块和我们BiSeNet的整个架构结合起来。

3.1语义分支

在语义分割的任务中，一些现有的方法[5, 6, 32, 40]试图保留输入图像的分辨率，用扩张卷积来编码足够的空间信息，而少数方法[5, 6, 26, 40]试图用金字塔集合模块、阿特拉斯空间金字塔集合或 "大核 "来捕捉足够的感受野。这些方法表明，空间信息和感受野是实现高精确度的关键。然而，要同时满足这两个要求是很难的。特别是在实时语义分割的情况下，现有的现代方法[1, 25, 39]利用小型输入图像或轻量级基础模型来加速。小尺寸的输入图像失去了原始图像的大部分空间信息，而轻量级的模型在通道修剪时损害了空间信息。基于这一观察，我们提出了一个空间路径，以保留原始输入图像的空间大小，并对富裕的空间信息进行编码。空间路径包含三层。每层包括一个跨度=2的卷积，然后是批量归一化[15]和ReLU[11]。因此，该路径提取的输出特征图是原始图像的1/8。由于特征图的空间尺寸较大，它编码了丰富的空间信息。图2(a)展示了结构的细节。

（绿色部分）  


3.2上下文分支

空间路径编码丰富的空间信息，而上下文路径则是为了提供足够的感受野。在语义分割任务中，接受域对性能具有重要意义。为了扩大感受野，一些方法利用了金字塔集合模块[40]、无轨空间金字塔集合[5,6]或 "大核"[26]。然而，这些操作对计算的要求很高，也很耗费内存，导致速度很低。考虑到大接收场和高效计算的同时，我们提出了 "上下文路径"。Context Path利用轻量级模型和全局平均池[5, 6, 21]来提供大的感受野。在这项工作中，轻量级模型，如Xception[8]，可以快速对特征图进行降样，以获得大的感受野，从而编码高层次的语义信息。然后，我们在轻量级模型的尾部添加一个全局平均池，它可以提供具有全局语境信息的最大感受野。最后，我们将全局池的上采样输出特征和轻量级模型的特征结合起来。在轻量级模型中，我们部署了U型结构[1, 25, 35]来融合后两个阶段的特征，这是一个不完整的U型结构风格。图2（c）显示了上下文路径的整体视角。注意力细化模块。在语境路径中，我们提出了一个特定的注意力细化模块（ARM）来细化每个阶段的特征。如图2(b)所示，ARM采用全局平均池来捕捉全局背景，并计算出一个注意力向量来指导特征学习。这种设计可以细化上下文路径中每个阶段的输出特征。它可以很容易地整合全局上下文信息，而不需要任何向上采样的操作。因此，它要求的计算成本可以忽略不计。  


3.3网络结构

通过空间路径和上下文路径，我们提出了BiSeNet，用于实时语义分割，如图2（a）所示。我们使用预先训练好的Xception模型作为 "语境路径 "的骨干，使用三个卷积层作为 "空间路径"。然后我们融合这两条路径的输出特征，进行最终的预测。它可以同时实现实时性能和高精确度。首先，我们把重点放在实际计算方面。虽然空间路径有很大的空间尺寸，但它只有三个卷积层。因此，它的计算量不大。至于 "上下文路径"，我们使用一个轻量级的模型来快速下采样。此外，这两条路径同时进行计算，这大大增加了效率。其次，我们讨论这个网络的准确性方面。在我们的论文中，空间路径编码了丰富的空间信息，而上下文路径提供了大的感受野。它们是相互补充的，以获得更高的性能。

特征融合模块。两条路径的特征在特征表示水平上是不同的。因此，我们不能简单地将这些特征相加。空间路径捕获的空间信息主要编码丰富的细节信息。此外，"语境路径 "的输出特征主要编码语境信息。换句话说，空间路径的输出特征是低层次的，而语境路径的输出特征是高层次的。因此，我们提出一个特定的特征融合模块来融合这些特征。考虑到特征的不同层次，我们首先将空间路径和上下文路径的输出特征连接起来。然后，我们利用批量归一化[15]来平衡这些特征的尺度。接下来，我们将串联的特征汇集成一个特征向量，并计算出一个权重向量，如SENet[13]。这个权重向量可以对特征进行重新加权，这相当于特征选择和组合。图2(c)显示了这个设计的细节。

