分支空洞卷积神经网络的机加工车间场景语义分割

刘强, 何自芬, 张印辉*

(昆明理工大学机电工程学院 昆明 650500)
(zhangyinhuai@kust.edu.cn)

摘要: 机加工车间场景的语义分割方法是开发工业场景中自主导航小车 AGV 所需要的一项关键技术. 针对 AGV 需要精确识别可通行区域和不可通行区域, 以及机加工车间场景中目标类别较多且密集导致难以准确分割的问题, 提出一种基于 DeepLabv3 深度学习模型架构的分支空洞卷积神经网络模型. 在预训练残差网络 ResNet-50 的基础上, 首先扩展分支结构, 通过分支结构设置不同比例的空洞卷积扩张率实现调节特征图感受野, 获取不同感受野的上下文信息; 然后通过相同扩张率的叠加状态改善空洞卷积的棋盘效应, 减少上下文信息的缺失; 最后添加多尺度特征融合的解码器单元, 利用目标定位准确的浅层特征和目标分类准确的深层特征进行特征融合, 弥补由于棋盘效应导致的上下文信息缺失和像素信息不相关性的问题. 在自制小样本机加工车间场景数据集上的实验结果表明, 与 DeepLabv3 模型相比, 该模型的验证精度提高 5.14%, 且对于可通行区域、道路线和不可通行区域的语义分割结果更加准确.

关键词: 语义分割; 空洞卷积; 机加工车间场景; 深度学习

中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2021.18383

Semantic Segmentation of Mechanical Workshop Scenes with Branch-Atrous Convolutional Neural Networks

Liu Qiang, He Zifen, and Zhang Yinhuai*

(Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500)

Abstract: The semantic segmentation method of mechanical workshop scene is a key technology required for the development of autonomous guided vehicles (AGV) in industrial scenes. Aiming at the problem that AGV needs to accurately identify the passable and impassable area, and the mechanical workshop scene has many and dense target categories, leading to the problem of difficult segmentation, a branch-atrous convolutional neural network model based on the DeepLabv3 deep learning network model architecture is proposed. On the basis of the pre-trained residual networks ResNet-50, the branch structure is first expanded, the atrous convolution with different expansion rates is set through the branch structure to achieve the adjustment of the feature map receptive field and obtain the context information of different receptive fields; then the gridding problem of the atrous convolution is improved through the superposition state of the same expansion rate, the lack of context information is reduced; finally, the model adds a decoder unit of multi-scale feature fusion, the decoder unit uses the shallow features with accurate positioning and the deep features with accurate classification to perform feature fusion, to compensate for the lack of context information and the irrelevance of pixel information due to the gridding problem. The experimental results on the self-made

收稿日期: 2020-05-11; 修回日期: 2020-11-06. 基金项目: 国家自然科学基金(61461022, 61761024, 62061022). 刘强(1993—), 男, 硕士研究生, 主要研究方向为计算机视觉、深度学习; 何自芬(1976—), 女, 博士, 副教授, 硕士生导师, 主要研究方向为图像处理、机器视觉、机器智能; 张印辉(1977—), 男, 博士, 教授, 博士生导师, 论文通讯作者, 主要研究方向为图像处理、机器视觉、机器智能.

small sample mechanical workshop scene dataset show that compared with the DeepLabv3 model, the verification accuracy of the model is improved by 5.14%, and the results of semantic segmentation for passable areas, road lines and impassable areas are more accurate.

Key words: semantic segmentation; atrous convolution; mechanical workshop scene; deep learning

现代工业生产自动化进程发展迅速, 自主导航小车(automated guided vehicle, AGV)广泛应用于柔性制造系统和智能仓储物流系统^[1]. 通过装配自动导引装置, 使具有装载功能的自主移动运输车能够沿规定的引导路径行驶, 或利用电学和光学导引装置规范其行驶路线.

与传统搬运车相比, AGV 具备行动灵活、使用便捷的特点. AGV 方法主要分为电磁导航^[2]、激光导航^[3]和视觉导航^[4-5]. 电磁导航路径设置柔性差, 成本高; 激光导航成本也相对较高, 其在密闭场所中噪声较大、易产生混叠且易丢失信号; 而视觉导航由于其不需要人为设置物理路径, 具有较好的引导柔性, 因此具有更广阔的研究价值.

作为计算机视觉的重要工作, 语义分割可以承担复杂多目标检测任务和场景理解任务. 在机加工车间场景中, AGV 需要识别技术工人和各类制造设备, 并根据不同功能区域区分可通行区域和不可通行区域、理解设备摆放位置和其他影响 AGV 自主移动的障碍物; 通过语义分割方法表达机加工车间场景中的可通行区域和多目标任务的区域标记和分类注释, 可进一步分析目标任务并理解场景.

在计算机视觉的工作中, 全卷积神经网络(fully convolutional networks, FCN)^[6], 最先实现端到端的训练. SegNet 模型^[7]通过引入更多的跳跃连接和最大池化指数, 使模型比 FCN 更为高效, 但是早期的语义分割模型产生的分割图较为粗略.

Yu 等^[8]在神经网络中提出空洞卷积结构, 在多尺度聚集条件下使用空洞卷积模块, 该模块由不同扩张率空洞卷积层联级产生, 在增大感受野的前提下, 改善池化操作导致的降低特征图像分辨率的问题.

Chen 等^[9]提出 DeepLabv1 模型, 通过利用空洞卷积结构在卷积操作过程中扩大感受野, 保持分辨率; 并通过增加全连接的条件随机场(conditional random field, CRF)使分割结果更精确.

Chen 等^[10]提出 DeepLabv2 模型, 通过引入空间金字塔池化(atrous spatial pyramid pooling, ASPP)结构, 以融合不同级别的语义信息. 为了能

够提取更复杂的特征, 上述语义分割模型构建了深层次网络结构, 结果导致明显的空间分辨率的下降.

Lin 等^[11]提出 RefineNet 模型, 通过使用残差卷积模块(residual convolution unit, RCU)^[12]和多分辨率融合模块, 实现融合高分辨率语义特征和低分辨率语义特征的模块, 以生成高分辨率的分割图像模型.

在语义分割方法上, 由于没有引入足够的上下文信息及不同感受野下的全局信息, 容易导致错误的分割结果. Zhao 等^[13]提出 PSPNet 模型, 使用金字塔池化模块(pyramid pooling module, PPM), 利用全局层次的特征信息对不同场景进行理解, 聚集不同区域的上下文信息以获取全局特征.

2018 年, Chen 等^[14]提出 DeepLabv3 模型放弃了 CRF 操作, 通过并行的、不同扩张率空洞卷积层, 并在每个并行卷积层之后加入批量归一化操作, 改进了 ASPP 模块; 为了防止空洞卷积感受野的扩张率过大导致的“权值退化”现象, 扩展了全局的平均池化卷积层, 利用全局信息增加全局特征.

上述模型均通过逐步增加网络深度, 以获取深层特征信息; 通过空洞卷积结构增大感受野, 使语义分割模型精度不断提升. 针对 AGV 在机加工车间场景中需要精确地识别可通行区域和不可通行区域, 其中目标类别较多且密集、难以准确地分割的问题, 本文提出一种基于 DeepLabv3 模型的分支空洞卷积神经网络(branch-atrous convolution neural networks, BACNet). 该模型扩展了分支结构, 通过设置不同比例的空洞卷积扩张率实现调节特征图感受野, 以获取不同相关性的上下文信息; 通过相同扩张率的叠加状态, 改善空洞卷积的棋盘效应, 减少上下文信息的缺失; 通过添加多尺度特征融合的解码器(decoder)单元, 利用目标定位相对准确的浅层特征和目标分类准确的深层特征进行特征融合, 弥补由于棋盘效应导致的上下文信息缺失和像素信息不相关性的问题. 实验结果表明, 本文模型比 DeepLabv3 模型具有更高的验证精度.

1 DeepLabv3 模型

DeepLabv3 模型将深度残差网络(deep residual

networks, ResNet)作为骨干网络, 在 ImageNet^[15]上进行预训练得到预训练参数, 模型结构示意图如图 1 所示.

图 1 DeepLabv3 模型结构

该骨干网络包含卷积层(Conv1)、池化层(Pool1)和 4 组卷积模块(Block)组成, 卷积模块分别称为 Block1, Block2, Block3 和 Block4 模块. 每个 Block 模块包含若干个残差单元, 每个残差单元包含瓶颈结构, 分别为 3 个卷积层 1×1, 3×3 和 1×1. 这种结构的中间层 3×3 的卷积首先通过一个 1×1 的卷积层降维, 最后通过一个 1×1 的卷积层进行维度还原, 既保持了精度又减少了计算量. DeepLabv3 模型对残差单元的中间层 3×3 的卷积层使用空洞卷积方法, 在防止深度网络退化的同时扩大感受野.

DeepLabv3 模型在 Block4 模块使用扩张率分别为 2, 4, 8 的空洞卷积, 以及 Block4 模块之后的 ASPP, 以实现多尺度特征的学习.

ASPP 使用扩张率分别为 6, 12, 18 的并行空洞卷积层进行多尺度特征处理, 通过增加平均池化操作来获取全局信息; 融合层(Fusion)对 ASPP 进行特征融合; 输出层(Logits)将输出特征维度减小为类别数量(Classes); 最后采用双线性插值上采样(Upsampling)还原图像尺寸进行分割预测. DeepLabv3 模型结构参数如表 1 所示.

2 分支空洞卷积神经网络模型

2.1 分支网络结构设计

将残差网络 ResNet-50 作为骨干网络, 在 ASPP 中扩展分支结构(branch structure, Branch), 分支结构包括 4 个分支路径, 分别在不同分支路径中添加 Block5 和 Block6 模块. 针对不同分支模块, 可以设置不同比例的空洞卷积扩张率, 通过大感

表 1 DeepLabv3 模型结构参数

网络结构	卷积核尺寸/步长	组数	特征图尺寸
Conv1	7×7×64/2	1	161×161
Pool1	3×3/2	1	81×81
Block1	1×1×64 3×3×64 1×1×256/2	3	41×41
Block2	1×1×128 3×3×128 1×1×512/2	4	21×21
Block3	1×1×256 3×3×256 1×1×1024	6	21×21
Block4	1×1×512 3×3×512 1×1×2048	3	21×21
ASPP	1×1×256 3×3×256 3×3×256 Pool+1×1×256	1	21×21
Fusion	1×1×256	1	21×21
Logits	1×1×Classes	1	21×21
Upsampling			321×321

受野卷积层检测分割较大的目标, 通过高分辨率特征定位目标类别, 以此定义不同分支结构.

对于使用空洞卷积的分支, 通过逐步扩大感受野, 以检测分割较大目标; 通过小感受野的分支检测分割较小目标; 对于不使用空洞卷积的分支, 通过全局池化精确定位目标. 通过调节特征图感受野获取不同感受野的上下文信息, 以提高对深层特征的感知; 通过 ASPP 扩张率为 6, 12, 18 的并行空洞卷积层进行特征处理和融合, 最后经过 1×1×256 的卷积层获取最终特征图像, 网络结构示意图如图 2 所示. 分支路径中 Block5 和 Block6 模块的结构参数如表 2 所示.

图 2 本文模型结构

表 2 分支路径中 Block5 和 Block6 模块的结构参数

模块	卷积核尺寸	特征图尺寸
Block5	1×1×512	21×21
	3×3×512	
	1×1×2048	
Block6	1×1×512	21×21
	3×3×512	
	1×1×2048	

在分支路径的 Block5 和 Block6 模块, 运用空洞卷积的相同扩张率叠加态可减轻棋盘效应, 增加上下文信息的相关性, 同时减少特征信息的丢失。

2.2 空洞卷积的相同扩张率叠加态

空洞卷积是指在普通卷积中添加零填充, 以扩展卷积核感受野的卷积方法, 图 3 所示为扩张率分别为 1, 2, 3 的空洞卷积。通过空洞卷积可以创建类似稀疏的滤波器。

图 3 不同扩张率的空洞卷积

虽然在不降低分辨率的前提下空洞卷积扩大感受野, 但是在卷积核 2 个采样像素间插入空值, 会导致棋盘效应问题。在某一特征图像中, 邻近的像素是从相互独立的子集中采样得到的, 因此导致特征图像局部信息丢失; 同时, 由于空洞卷积固有的稀疏采样方法, 使卷积得到的信息缺少相关性, 影响分类结果。

空洞卷积的卷积核尺寸的计算公式为

$$N = [n + (n-1) \times (r-1)] \quad (1)$$

其中, n 为普通卷积核尺寸; r 为空洞卷积扩张率; N 为空洞卷积的卷积核尺寸。

经过空洞卷积的特征图尺寸的计算公式为

$$W = \frac{w + 2p - r \times (n-1) - 1}{s} + 1 \quad (2)$$

其中, w 为输入图像的尺寸; n 为普通卷积核尺寸; p 为填充补零层数; s 为卷积步长; W 为输出特征图的尺寸。

在卷积神经网络结构中, 卷积层输出的每个像素信息仅依赖该卷积层输入的一个区域, 这个区域之外的其他输入的信息都不会影响该像素。通过普通卷积操作, 在不进行补零填充或步长为 2 的条件下, 所得到的特征图变为原始图像的 1/2; 随着不断进行卷积操作, 特征图分辨率不断降低, 在最终特征图中, 某一个像素点在原始图像上表示的像素范围称为感受野, 该像素点能感知原始图像上某个区域的大小, 可以反映该区域的特征信息, 即是否存在目标类别。

相对于普通卷积而言, 空洞卷积的优势在于不改变特征图分辨率的前提下, 感受野比普通卷积更大, 感知信息的范围更大。

计算感受野尺寸, 即计算卷积网络每一层输出特征图上的像素点在原始图像上映射区域的大小。假设不进行填充补零操作, 求解感受野的尺寸计算公式为

$$y_k = y_{k-1} + (n_k - 1) \times \prod_{i=1}^{k-1} s \quad (3)$$

其中, k 表示第 k 层卷积层; y_k 表示本层感受野尺寸; y_{k-1} 表示前一层感受野尺寸。

将普通卷积替换为空洞卷积时, 假设不进行填充补零操作, 计算公式为

$$y_k = y_{k-1} + [n_k + (n_k - 1) \times (r_k - 1) - 1] \times \prod_{i=1}^{k-1} s \quad (4)$$

对其简化后可得

$$y_k = y_{k-1} + [r_k \times (n_k - 1)] \times \prod_{i=1}^{k-1} s \quad (5)$$

若 $s=1$, 则式(5)可以表示为

$$y_k = y_{k-1} + [r_k \times (n_k - 1)] \quad (6)$$

可以假设从最终特征图的一个像素点出发, 逆推前一层空洞卷积的感受野, 其计算公式为

$$y_k = r_k \times (n_k - 1) + 1 \quad (7)$$

其中, y_k 表示最终特征图的一个像素点在前一层特征图上的感受野尺寸; n_k 表示卷积核尺寸; r_k 表示空洞卷积扩张率, 即最终特征图的一个像素点在前一层特征图上的感受野尺寸等于经过的空洞卷积的卷积核尺寸。

前一个层的感受野尺寸, 计算公式为

$$y_{k-1} = y_k + r_{k-1} \times (n_{k-1} - 1) \quad (8)$$

假设某层特征图再经过 3 层卷积层后得到最终特征图, 3 层卷积层均为 3×3 卷积, 且 $s=1$ 时, 则式(7)可以简化为

$$y_3 = 2 \times r_3 + 1 \quad (9)$$

其中, y_3 表示第 3 层的感受野尺寸; r_3 表示第 3 层的空洞卷积扩张率。

式(8)可以简化为

$$\begin{cases} y_2 = y_3 + 2 \times r_2 \\ y_1 = y_2 + 2 \times r_1 \end{cases} \quad (10)$$

其中, y_2 和 y_1 分别表示第 2 层和第 1 层的感受野尺寸; r_2 和 r_1 分别表示第 2 层和第 1 层的空洞卷积扩张率。

根据式(7)~(10)可以计算多层不同空洞卷积扩张率组合的感受野的变化情况. 以 DeepLabv3 模型的 Block4 模块为例, 假设从当前特征图的一个像素点(红色)出发, 逆向推导空洞卷积扩张率为 2, 4, 8 时感受野的变化情况如图 4 所示, 前 3 层的感受野分别为 17×17 , 25×25 和 29×29 。

图 4 空洞卷积扩张率为 2, 4, 8 的感受野变化

可以看出扩张率组合的公约数越大, 信息缺失情况越明显, 卷积神经网络提取深层特征时, 对目标的识别和定位效果也越差. 当扩张率为 3, 6, 9 时, 采样像素之间会出现 2 个像素的空洞; 当扩张率为 4, 8, 12 时, 采样像素之间会出现 3 个像素的空洞. 随着空洞的增大, 采样像素之间缺失信息增

多, 像素的相关性大幅减小, 不利于目标定位。

假设从当前特征图的一个像素点(红色)出发, 逆向推导前 3 层的感受野尺寸变化, 常见的扩张率组合的感受野如表 3 所示. 可以看出, 映射到第 1 层的感受野尺寸为 19×19 、扩张率组合分别是 1, 3, 5 和 2, 3, 4 时, 将上述感受野变化情况通过棋盘图像表示如表 4 所示, 在第 1 层感受野相同的情况下, 扩张率为 1, 3, 5 的效果更好。

表 3 不同空洞卷积扩张率逆向推导前 3 层感受野

扩张率	第 3 层	第 2 层	第 1 层
1, 3, 5	11×11	17×17	19×19
2, 3, 4	9×9	15×15	19×19
2, 4, 6	13×13	21×21	25×25
3, 4, 5	11×11	19×19	25×25
4, 5, 6	13×13	23×23	31×31
3, 5, 7	15×15	25×25	31×31

表 4 感受野尺寸 19×19 的扩张率组合对比

扩张率	第 3 层	第 2 层	第 1 层
1, 3, 5			
2, 3, 4			

同理, 映射到第 1 层的感受野尺寸为 25×25 、扩张率组合为 2, 4, 6 和 3, 4, 5 时, 感受野变化情况通过棋盘图像表示如表 5 所示. 可以看出, 扩张率为 2, 4, 6 的全偶数情况下上下文信息缺失较严重, 容易造成像素信息相关性降低。

表 5 感受野尺寸 25×25 的扩张率组合对比

扩张率	第 3 层	第 2 层	第 1 层
2, 4, 6			
3, 4, 5			

当映射到第 1 层的感受野尺寸为 31×31 、扩张率组合为 3, 5, 7 和 4, 5, 6 时, 如表 6 所示. 可以看出, 全奇数扩张率和奇偶混合扩张率在第 1 层感受

野相同的情况下, 对于中心部分的目标的棋盘效应较轻, 边缘部分容易出现信息丢失; 但相比全偶数扩张率, 棋盘效应明显减轻, 信息相关性损失较小, 图像分割效果更好。

表6 感受野尺寸 31×31 的扩张率组合对比

扩张率	第3层	第2层	第1层
3, 5, 7
4, 5, 6

由于感受野不是无限增大的, 当某一层的感受野大于特征图尺寸时, 映射到原始图像的感受野将大于原始图像的尺寸; 此时空洞卷积将会退化为 1×1 的普通卷积, 将会引入额外计算量, 增加计算成本, 同时对图像边缘检测效果将会降低。因此, 通过计算某一层特征图尺寸可以为选取空洞卷积扩张率提供依据, 在感受野一定的情况下, 通过选取合理的扩张率组合, 可以有效地降低棋盘效应。

相同空洞卷积扩张率的感受野变化情况如表7所示, 图5所示为表7感受野变化情况通过棋盘图像表示。

表7 相同空洞卷积扩张率逆向推导前3层感受野

扩张率	第3层	第2层	第1层
1, 1, 1	3×3	5×5	7×7
2, 2, 2	5×5	9×9	13×13
3, 3, 3	7×7	13×13	19×19

图5 相同空洞卷积扩张率叠加的感受野变化

由图5可以看出, 随着扩张率的增大, 空洞越多, 采样像素间信息丢失越多, 通过叠加较小扩张率空洞卷积, 可以有效地填充采样像素的间隙, 弥补棋盘效应。例如, 通过叠加扩张率为 2, 2, 2+3, 3, 3 时, 可以计算当前特征图的一个像素逆推前6层的感受野变化, 如表8所示。

表8 扩张率为 2, 2, 2+3, 3, 3 的感受野变化

卷积层	感受野
第6层	7×7
第5层	13×13
第4层	19×19
第3层	23×23
第2层	27×27
第1层	31×31

通过对比相同感受野的扩张率组合, 如图6所示, 逆推第1层的感受野为 31×31 , 可以看出, 扩张率组合为 2, 2, 2+3, 3, 3 时棋盘效应明显减小, 边缘的信息丢失明显减少。同理, 如图7所示, 扩张率组合为 1, 1, 1+2, 2, 2 的相同扩张率叠加态, 边缘的信息丢失也有所减少。

图6 第1层感受野为 31×31 的空洞卷积扩张率组合

图7 第1层感受野为 19×19 的空洞卷积扩张率组合

通过计算可以得出扩张率组合为 3, 3, 3+4, 4, 4 的感受野变化情况从第6层逆推至第1层, 感受野分别为 9×9 , 17×17 , 25×25 , 31×31 , 37×37 和 43×43 。在感受野持续扩大的情况下, 如图8所示, 棋盘效应明显改善, 但是随着扩张率的增大, 边缘的信息丢失情况依然存在。

图8 扩张率为 3, 3, 3+4, 4, 4 逆推至第1层的感受野

2.3 多尺度特征融合解码器

对于 DeepLabv3 模型, 通过骨干网络和 ASPP 的特征图像在深度特征提取方面取得较好的结果, 但是 DeepLabv3 模型未使用解码器模块, 而是通过双线性插值上采样方法将尺寸为 21×21 的最终特征图还原尺寸为 321×321 的预测图像。

由于模型使用了空洞卷积, 在采样像素之间插入空值容易造成像素信息丢失, 同时造成像素之间相关性降低。虽然双线性插值的方法可以补充采样像素间丢失的部分像素信息, 但是像素之间相关性减弱, 容易造成目标类别的边缘信息模糊, 对于小目标容易出现分割错误的现象。

因此, 可以通过添加多尺度特征融合的解码器模块还原图像尺寸, 利用高分辨率、目标定位准确的浅层特征和低分辨率、目标分类准确的深层特征进行特征融合, 实现融合高分辨率语义特征和低分辨率语义特征的解码器模块, 以生成高分辨

率分割模型的预测图像。

本文对比 2 类多尺度特征融合解码器, 如图 9 所示, 通过提取不同尺度的特征图像, 实现逐层融合特征, 以弥补经过不同 Block 模块导致的空洞卷积信息丢失的问题, 同时有效地运用 Block 模块输出的特征图信息。如图 10 所示, 此类解码器将 ASPP 输出的深度特征信息和一层浅层特征信息进行特征融合, 由于浅层特征包含准确的定位信息, 因此使用浅层特征对缺失像素进行填充, 以增加采样像素之间的相关性。

通过不同 Block 模块输出的特征图, 采用逐像素相加的方式进行特征融合, 再通过感受野为 3×3 , 步长为 1 的卷积层输出图像。通过结合浅层特征图像, 可以使最终得到的特征图中既包含深层的高级语义信息, 也包含丰富的浅层的低级语义信息, 实现了多尺度特征融合的功能, 提高模型的训练精确度。

图 9 逐层特征融合的解码器结构

图 10 浅层特征和深层特征融合的解码器结构

2.4 评价指标

本文采用平均交并比(mean intersection over union, mIoU)作为网络性能的核心评价指标, 并使用检测时间作为辅助评价指标。

交并比(intersection over union, IoU)是某个类别的预测的结果和该类别的真实值的交集与并集的比值。利用混淆矩阵表示 IoU 包含真正例(true positive, TP), 即实际是目标, 预测也是目标; 假正例(false positive, FP), 即实际不是目标, 但预测成目标; 假负例(false negative, FN), 即实际不是目标, 预测也不是目标。IoU 计算公式为

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (11)$$

mIoU 就是所有类别的 IoU 的平均值, 其公式为

$$\text{mIoU} = \frac{1}{n+1} \frac{\sum_{i=0}^n p_{ii}}{\sum_{j=0}^n p_{ij} + \sum_{j=0}^n (p_{ji} - p_{ii})} \quad (12)$$

其中, n 表示类别数量, 共有 $n+1$ 个类别, 包括一个背景类别; p_{ij} 表示属于第 i 类, 但被预测为第 j 类的像素数量, 即真负例; p_{ii} 表示属于第 i 类, 同时预测为第 i 类的像素数量, 即真正例; 同理 p_{ji} 表示为假正例。

3 实验结果及分析

3.1 实验数据集

本文使用 Windows 10 64 位操作系统, 硬件环境为 Intel(R) Core(TM) i5-8400 CPU @ 2.80 GHz 处理器, 8GB 内存, GPU 为 NVIDIA GTX 1060 Ti 6GB 的笔记本电脑; 深度学习框架为 TensorFlow 1.8.0。

本文通过昆明理工大学机械加工作坊场景环境获取 235 幅实拍图像, 通过网上下载 365 幅机加工车间场景图像作为数据集扩充, 共形成 600 幅图像。统一将图像尺寸修改为 321×321 , 通过人工标注对应图像的真实值(ground truth, GT)获得图像数据集。

该图像数据集划分为训练集 500 幅, 验证集 80 幅和测试集 20 幅, 各数据集之间图像不重复, 数据集共包括 15 类分类目标和 1 类图像背景。

该数据集包括不可通行区域、可通行区域, 道路线、窗户、门、普通机床、数控机床、加工中心、桥式起重机、人、桌子、椅子、电脑、货柜和零件或零件箱 15 类目标, 像素值如表 9 所示。

表 9 机加工车间场景对应各类别 GT 像素值

序号	目标	像素值
1	不可通行区域	(128, 0, 0)
2	可通行区域	(0, 128, 0)
3	道路线	(128, 128, 0)
4	窗户	(0, 0, 128)
5	门	(128, 0, 128)
6	普通机床	(0, 128, 128)
7	数控机床	(128, 128, 128)
8	加工中心	(64, 0, 0)
9	桥式起重机	(192, 0, 0)
10	人	(64, 128, 0)
11	桌子	(192, 128, 0)
12	椅子或凳子	(64, 0, 128)
13	电脑	(192, 0, 128)
14	货柜	(64, 128, 128)
15	零件	(192, 128, 128)
0	背景墙面	(0, 0, 0)

3.2 训练参数选取与优化器

为了确定最优训练参数, 本文采用 DeepLabv3 模型作为基准模型, 通过单一变量法分别调整训练步数、批尺寸和学习率。

设置批尺寸为 4, 学习率为 0.001 00 时, 调整训练步数, 从 10 000 步到 30 000 步, 每组间隔 2 500 步, 实验结果如图 11 所示, 训练步数为 17 500 时, 验证精度最高达到 49.20%。

图 11 不同训练步数的训练和验证精度

设置学习率为 0.001 00, 训练步数为 17 500 步, 调整批尺寸, 批尺寸从 2 到 10, 每组间隔为 2, 实验结果如图 12 所示, 批尺寸为 4 时, 验证精度最高。

设置批尺寸为 4, 训练步数为 17 500 步, 调整学习率从 0.010 00~0.000 01。实验结果如表 10 所示, 学习率为 0.010 00 时, 验证精度最高, 达到 51.28%。

由此确定最佳训练参数, 批尺寸为 4, 学习率

图 12 不同批尺寸的训练和验证精度

表 10 不同学习率的训练和验证精度 %

学习率	mIoU	
	训练	验证
0.010 00	76.40	51.28
0.005 00	72.63	50.61
0.001 00	60.74	49.20
0.000 50	52.68	44.79
0.000 10	32.24	33.51
0.000 05	24.76	26.60
0.000 01	14.86	17.58

为 0.010 00, 训练步数为 17 500 步。

通过训练参数选取实验获得适合于该模型的最优的参数设定, 使用 ImageNet 预训练的 ResNet 骨干网络, 其他参数和优化器如表 11 所示。

表 11 其他参数和优化器

设置项	参数
优化器	Mini-batch SGD
输入图像尺寸	321×321
学习率策略	Poly, Power=0.9
权重衰减	0.000 1
动量	0.9

3.3 扩展分支结构实验分析

本组实验使用残差网络 ResNet-50 作为骨干网络, 在 ASPP 中添加 4 组分支 Block 模块, 通过调整不同分支组别的空洞卷积扩张率组合, 验证分支结构效果。

添加分支结构 Block5 模块, 4 组分支结构采用相同的空洞卷积扩张率组合, 实验结果如表 12 所示。使用普通卷积时, 验证精度提升 2.00%, 但是随着扩张率组合的增大精度基本保持不变, 这是因为空洞卷积的感受野已经超过了图像尺寸, 退化为 1×1 卷积, 因此无法感知更大的特征区域。

表 12 添加相同扩张率的 Block5 模块的精度 %

扩张率	mIoU	
	训练	验证
	76.40	51.28
1, 1, 1	76.34	53.28
1, 2, 4	75.11	49.70
2, 4, 8	74.45	51.20
4, 8, 16	74.84	51.87
8, 16, 32	75.05	51.26

对 Block5 模块的 4 组分支结构采用不同的空洞卷积扩张率组合, 实验结果如表 13 所示。当第 1 组扩张率设置为 1, 2, 4, 第 2~4 组扩张率设置为 2, 4, 8 时, 验证精度提升 2.40%, 其他组合精度未明显提升或保持不变。

表 13 添加不同扩张率的 Block5 模块的精度 %

扩张率		mIoU	
第 1 组	第 2~4 组	训练	验证
		76.40	51.28
1, 1, 1	1, 2, 4	75.77	51.24
1, 1, 1	2, 4, 8	76.05	51.52
1, 1, 1	4, 8, 16	76.52	51.17
1, 2, 4	1, 1, 1	75.66	51.39
2, 4, 8	1, 1, 1	75.68	51.59
4, 8, 16	1, 1, 1	75.34	50.81
1, 2, 4	2, 4, 8	75.54	53.68
2, 4, 8	4, 8, 16	75.54	51.00
2, 4, 8	1, 2, 4	74.54	51.66
4, 8, 16	2, 4, 8	74.79	49.81

3.4 空洞卷积的相同扩张率叠加态分析

由于本组实验目的是验证相同扩张率叠加态对模型的分割能力, 因此简化模型结构, 以消除其他结构对结果的影响。

本组实验使用 ResNet-50 作为骨干网络, 在 Block4 模块之后添加 Block5 和 Block6 模块, Block6 模块连接 ASPP, 最后通过双线性插值上采样获得预测图像。通过调节 Block5 和 Block6 模块的空洞卷积扩张率, 实现相同扩张率叠加态方法。

通过计算可以得到相同扩张率叠加态的感受野变化情况, 假设从 Block6 的特征图的一个像素点出发, 通过 3×3 卷积, 步长为 1 时, 逆推 6 层空洞卷积的感受野变化情况如表 14 所示。

实验结果如表 15 所示, 当扩张率为 1, 1, 1+1, 1, 1 时, 不使用空洞卷积。当扩张率为 1, 1, 1+2, 2,

2 时, 出现相同扩张率叠加态, 此时验证精度为 53.17%, 与不使用空洞卷积结果相比, 精度提升了 0.81%。当扩张率为 2, 2, 2+3, 3, 3 或更大时, 验证精度保持不变或下降。

表 14 Block5+Block6 相同扩张率叠加态感受野变化

扩张率	卷积层					
	6	5	4	3	2	1
1, 1, 1+1, 1, 1	3×3	5×5	7×7	9×9	11×11	13×13
1, 1, 1+2, 2, 2	5×5	9×9	13×13	15×15	17×17	19×19
2, 2, 2+3, 3, 3	7×7	13×13	19×19	23×23	27×27	31×31
3, 3, 3+4, 4, 4	9×9	17×17	25×25	31×31	37×37	43×43
4, 4, 4+5, 5, 5	11×11	21×21	31×31	39×39	47×47	55×55
5, 5, 5+6, 6, 6	13×13	25×25	37×37	47×47	57×57	67×67

表 15 Block5+Block6 相同扩张率叠加态的精度 %

扩张率	mIoU	
	训练	验证
1, 1, 1+1, 1, 1	76.04	52.36
1, 1, 1+2, 2, 2	75.59	53.17
2, 2, 2+3, 3, 3	73.93	50.31
3, 3, 3+4, 4, 4	74.39	51.08
4, 4, 4+5, 5, 5	74.23	51.13
5, 5, 5+6, 6, 6	74.12	51.33

分析原因, 通过表 14 可以看出, 随着扩张率的增大, 感受野不断提升, 当扩张率为 2, 2, 2+3, 3, 3 时, 第 1 层感受野为 31×31, 此时感受野超过了特征图尺寸 21×21, 即超过了特征图像的分辨率。由此逆推至原始图像, 其感受野映射在原始图像时已经超过 321×321, 即空洞卷积退化 1×1 卷积, 因此无法感知更大的特征区域。同时, 由于添加了 Block5 和 Block6 模块, 网络深度大幅增加, 导致训练过拟合, 因此之后的相同扩张率叠加态在实验中表现出精度下降或不变的情况。

为了验证相同扩张率叠加态对模型的影响, 增大 Block4 输出的特征图尺寸, 以匹配更大的相同扩张率叠加态的感受野。

此时将输入图像尺寸修改为 513×513, 使用 ResNet-50 作为骨干网络, 在 Block4 模块后添加 Block5 和 Block6 模块, Block6 模块连接 ASPP, 最终输出的特征图通过双线性插值上采样还原成原始图像的分辨率, 获得预测图像, 实验结果如表 16 所示。

从表 16 可以看出, 扩张率为 1, 1, 1+1, 1, 1, 不使用空洞卷积, 精度为 53.67%; 使用相同扩张率叠加态之后, 验证精度分别提升了 0.74%, 0.63%, 1.13% 和 0.92%。

表 16 输入图像 513×513 时相同扩张率叠加态精度 %

扩张率	mIoU	
	训练	验证
1, 1, 1+1, 1, 1	81.14	53.67
1, 1, 1+2, 2, 2	79.88	54.41
2, 2, 2+3, 3, 3	79.26	54.30
3, 3, 3+4, 4, 4	79.07	54.80
4, 4, 4+5, 5, 5	78.88	54.59

最后, 输入图像尺寸依然保持 513×513, 使用 ResNet-50 作为骨干网络, 删除 Block6 模块, 通过修改 Block4 和 Block5 模块的空洞卷积扩张率组合实现相同扩张率叠加态, 实验结果如表 17 所示。

表 17 Block4+Block5 相同扩张率叠加态的精度 %

扩张率	mIoU	
	训练	验证
1, 1, 1+1, 1, 1	80.47	54.18
1, 1, 1+2, 2, 2	80.15	54.83
2, 2, 2+3, 3, 3	79.99	53.90
3, 3, 3+4, 4, 4	80.00	55.72
4, 4, 4+5, 5, 5	79.24	54.40

从表 17 可以看出, 使用相同扩张率叠加态验证精度最高达到 55.72%, 与扩张率为 1, 1, 1+1, 1, 1 的不使用空洞卷积的模型结果相比, 精度提升了 1.54%。

整理上述对比结果如表 18 所示。可以看出, 与不使用空洞卷积模型相比, 叠加态方法的精度分别提升 0.81%, 1.13% 和 1.54%; 增大输入图像尺寸, 模型获得更大特征图后, 相同扩张率叠加态方法精度提升 1.63%; 增大输入图像尺寸并减小模型深度后, 验证精度提升 2.55%。实验结果表明, 相同扩张率叠加态方法可以提升模型精度, 同时模型深度减小也可以获得精度的提升。

表 18 相同扩张率叠加态方法的验证精度对比 %

输入尺寸	Block6	普通卷积	叠加态方法
321×321	√	52.36	53.17
513×513	√	53.67	54.80
513×513		54.18	55.72

3.5 混合空洞卷积与本文模型的实验比较

本组实验的目的是比较混合空洞卷积(hybrid dilated convolution, HDC)^[16]和空洞卷积的相同扩张率叠加态方法在机加工车间场景数据集的精度。

HDC 通过定义 2 个采样像素的最大距离, 计

算采样像素之间的丢失像素的数量, 设置上一层的空洞卷积扩张率, 以填充损失的像素区域, 从而使感受野全覆盖特征图像, 减小空间和边缘的信息损失, 并给出一种选择空洞卷积扩张率的设计原则. HDC 的计算公式为

$$\begin{cases} M_i = \max[M_{i+1} - 2 \times r_i, M_{i+1} - 2 \times (M_{i+1} - r_i), r_i] \\ M_k = r_k \\ M_2 \leq n \end{cases} \quad (13)$$

其中, M_i 表示第 i 层使用空洞卷积时 2 个采样像素之间的最大距离; r_i 表示第 i 层使用空洞卷积扩张率; M_k 表示最后一层的采样像素之间的最大距离; r_k 表示最后一层空洞卷积的扩张率, 即最后一层使用空洞卷积时采样像素之间的最大距离等于空洞卷积的扩张率; n 表示原始卷积的卷积核尺寸. 设计目标为第 2 层的 2 个采样像素之间的最大距离小于或等于原始卷积核的尺寸, 即可以通过 3×3 卷积填充空洞卷积导致的像素缺失.

验证扩张率组合是否满足 HDC 方法如表 19 所示. 选取部分扩张率组合, 假设从 Block6 模块特征图的一个像素点出发, 通过 3×3 卷积, 步长为 1 时, 逆推 6 层空洞卷积的感受野变化情况, 如表 20 所示.

实验使用 ResNet-50 作为骨干网络, 在 Block4 模块之后添加 Block5 和 Block6 模块, Block6 模块连接 ASPP, 最后通过双线性插值上采样获得预测图像. Block4 模块空洞卷积扩张率为 2, 4, 8, 通过调节 Block5 和 Block6 模块的空洞卷积扩张率实现 HDC, 实验结果如表 21 所示.

表 19 HDC 的空洞卷积扩张率

扩张率	$M_3 - 2r_2$	$-M_3 + 2r_2$	r_2	M_2	$M_2 \leq n$
1, 2, 3	-1	1	2	2	√
1, 2, 5	1	-1	2	2	√
1, 2, 6	2	-2	2	2	√
1, 3, 6	0	0	3	3	√
1, 3, 7	1	-1	3	3	√
1, 3, 8	2	-2	3	3	√
1, 3, 9	3	-3	3	3	√
1, 4, 6	-2	2	4	4	
1, 4, 7	-1	1	4	4	
1, 5, 9	-1	1	5	5	
2, 2, 3	-1	1	2	2	√
2, 2, 5	1	-1	2	2	√
2, 3, 7	1	-1	3	3	√
2, 3, 9	3	-3	3	3	√

表 20 不同卷积下 HDC 的空洞卷积扩张率感受野

扩张率	卷积层					
	6	5	4	3	2	1
1, 2, 3+1, 2, 3	7×7	11×11	13×13	19×19	23×23	25×25
1, 2, 5+1, 2, 5	11×11	15×15	17×17	27×27	31×31	33×33
1, 3, 7+1, 3, 7	15×15	21×21	23×23	37×37	43×43	45×45
1, 3, 9+1, 3, 9	19×19	25×25	27×27	45×45	51×51	53×53
2, 2, 3+2, 2, 3	7×7	11×11	15×15	21×21	25×25	29×29
2, 2, 5+2, 2, 5	11×11	15×15	19×19	29×29	33×33	37×37
2, 3, 7+2, 3, 7	15×15	21×21	25×25	39×39	45×45	49×49
2, 3, 9+2, 3, 9	19×19	25×25	29×29	47×47	53×53	57×57

表 21 调节 Block5 和 Block6 模块后 HDC 的精度 %

扩张率	mIoU	
	训练	验证
1, 1, 1+1, 1, 1	76.04	52.36
1, 2, 3+1, 2, 3	75.39	53.41
1, 2, 5+1, 2, 5	75.03	52.71
1, 3, 7+1, 3, 7	75.60	52.92
1, 3, 9+1, 3, 9	74.93	51.33
2, 2, 3+2, 2, 3	74.97	49.56
2, 2, 5+2, 2, 5	75.06	50.33
2, 3, 7+2, 3, 7	74.37	50.99
2, 3, 9+2, 3, 9	75.22	49.99

通过表 21 可以看出, 当扩张率为 1, 1, 1+1, 1, 1 时, 不使用空洞卷积, 仅为增加模型深度的结果; 根据选取的 HDC 的空洞卷积扩张率, 精度最高达到 53.41%, 比不使用空洞卷积结果提升了 1.05%.

通过实验发现, 当第 1 层的扩张率不等于 1 时, 如等于 2, 精度明显下降; 分析原因为当第 1 层的扩张率不等于 1 时, 边缘信息依然出现像素信息丢失的问题, 红色像素表示当前特征图的一个像素点, 分别逆推前 3 层的棋盘像素变化情况如图 13 所示.

对比图 13a 和图 13d 可以看出, 当第 1 层的扩张率不等于 1 时, 边缘像素采样时也会出现少量像素信息缺失; 对比图 13b 和图 13e, 图 13c 和图 13f 可以看出, 随着扩张率的增大, 这种情况依然存在, 边缘像素信息无法得到有效的补充.

使用 HDC 后, 红色像素表示当前特征图的一个像素点, 分别逆推前 6 层的感受野变化情况如图 14 所示.

对比图 14a 和图 14b 可以看出, 当第 1 层的扩张率等于 1 时, 通过 3×3 卷积核可以填充空洞卷积造成的采样像素之间丢失的信息; 但当第 1 层的扩张率不等于 1 时, 即使使用 HDC, 边缘像素信息依

图 13 逆推前 3 层的 HDC 像素丢失

图 14 逆推前 6 层的 HDC 像素丢失

然会出现少量丢失情况。

为了对比相同扩张率叠加态的实验结果, 使用 513×513 的输入图像尺寸, 实验结果如表 22 所示。此时 HDC 的最高精度达到了 55.95%, 比扩张率为 1, 1, 1+1, 1, 1 的不使用空洞卷积的精度提升了 2.28%。

表 22 输入图像 513×513 时 HDC 的精度 %

扩张率	mIoU	
	训练	验证
1, 1, 1+1, 1, 1	81.14	53.67
1, 2, 3+1, 2, 3	79.95	55.74
1, 2, 5+1, 2, 5	79.81	55.04
1, 3, 7+1, 3, 7	79.78	55.95
1, 3, 9+1, 3, 9	80.23	54.56

最后, 输入图像尺寸仍为 513×513 , 减小模型深度, 删除 Block6 模块, 通过调节 Block4 和 Block5 模块的扩张率组合实现 HDC。实验结果如表 23 所示, HDC 精度最高达到 56.50%, 相比扩张率为 1, 1, 1+1, 1, 1 的不使用空洞卷积的精度提升了 2.32%。

对比上述实验, 叠加态方法和 HDC 的验证精度结果如表 24 所示。

HDC 与叠加态方法相比, 当输入图像尺寸为 321×321 时, 其验证精度高 0.24%; 当输入图像尺

表 23 调节 Block4+Block5 模块后 HDC 的精度 %

扩张率	mIoU	
	训练	验证
1, 1, 1+1, 1, 1	80.47	54.18
1, 2, 3+1, 2, 3	80.49	54.30
1, 2, 5+1, 2, 5	80.70	55.11
1, 3, 7+1, 3, 7	79.94	56.50
1, 3, 9+1, 3, 9	80.31	54.57

表 24 叠加态方法和 HDC 验证精度对比 %

输入尺寸	Block6	叠加态方法	HDC ^[16]
321×321	√	53.17	53.41
513×513	√	54.80	55.95
513×513		55.72	56.50

寸为 513×513 时, 其验证证精度高 1.15%; 减小模型深度后, 其验证精度高 0.78%。从表 24 可以看出, 当输入图像尺寸较小、模型深度较深时, 2 种方法结果相近; 当输入图像尺寸较大、模型深度较浅时, HDC 效果更好; 其在空洞卷积层较少时也能表现出较好的结果。因此, 在选取方法时需要综合考虑特征图尺寸和模型深度。

3.6 简化后的相同扩张率叠加态实验分析

当输入图像尺寸为 321×321 时, 原相同扩张率叠加态为 6 层。若感受野过大, 会出现空洞卷积退化问题; 而模型深度加深, 则会计算量增大。因此, 修改 Block5 和 Block6 模块分别进行 2 组空洞卷积, 将相同扩张率叠加态简化为 4 层, 以降低叠加态在第 1 层的感受野尺寸, 同时减小模型深度, 防止训练过拟合。

假设当前特征图的一个像素点用红色像素表示, 将简化后的相同扩张率叠加态分别逆推至前 4 层的棋盘像素变化情况, 如图 15 所示, 可以看出, 简化后的相同扩张率叠加态依然能减少采样像素丢失的问题, 中心区域得到了有效的填充; 但是随着扩张率的增大, 边缘像素信息逐渐缺失。

图 15 简化后的相同扩张率叠加态感受野变化情况

通过计算可以得到简化后的相同扩张率的空洞卷积叠加态的感受野变化情况。假设从 Block6

的特征图的一个像素点出发,通过 3×3 卷积,步长为 1 时,逆推 4 层空洞卷积的感受野变化情况如表 25 所示。

表 25 简化后的相同扩张率叠加态感受野变化情况

扩张率	卷积层			
	4	3	2	1
1, 1+1, 1	3×3	5×5	7×7	9×9
1, 1+2, 2	5×5	9×9	11×11	13×13
2, 2+3, 3	7×7	13×13	17×17	21×21
3, 3+4, 4	9×9	17×17	23×23	29×29
4, 4+5, 5	11×11	21×21	29×29	37×37

实验结果如表 26 所示,扩张率为 1, 1+1, 1 时,表示未使用空洞卷积的实验结果;当扩张率为 2, 2+3, 3 时,精度达到 52.32%。此时第 1 层的感受野尺寸为 21×21 ,等于特征图尺寸,模型精度提升;当扩张率为 3, 3+4, 4 和 4, 4+5, 5 时,第 1 层的感受野尺寸分别为 29×29 和 37×37 ,感受野尺寸超过特征图尺寸,空洞卷积退化,此时模型的训练和验证精度明显下降。

表 26 简化后的相同扩张率叠加态的精度 %

扩张率	mIoU	
	训练	验证
1, 1+1, 1	76.20	51.66
1, 1+2, 2	76.10	51.19
2, 2+3, 3	74.54	52.32
3, 3+4, 4	74.28	50.28
4, 4+5, 5	74.63	50.35

3.7 分支结构组合相同扩张率叠加态实验分析

为了验证分支结构组合相同扩张率叠加态可以提升模型效果,本组实验使用 ResNet-50 作为骨干网络,在 ASPP 中添加 4 组分支结构,每组分支结构包含 Block5 和 Block6 模块。通过调节 Block5 和 Block6 模块的空洞卷积扩张率组合,实现相同扩张率叠加态方法。

当输入图像尺寸为 321×321 时,由于模型过深,计算量增大,同时感受野过大会出现空洞卷积退化问题。因此,修改 Block5 和 Block6 模块,分别进行 2 组空洞卷积,使用简化后的相同扩张率叠加态,实验结果如表 27 所示。分支结构组合简化后的相同扩张率叠加态,模型精度最高达到 51.63%。

当扩大输入图像尺寸为 513×513 时,使用 6 层相同扩张率叠加态,还原 Block5 和 Block6 模块分别进行 3 组空洞卷积,实验结果如表 28 所示。

表 27 输入图像尺寸 321×321 时本文模型的精度 %

第 1 组	第 2~4 组	mIoU	
		训练	验证
1, 1+1, 1	1, 1+1, 1	73.80	49.18
1, 1+2, 2	1, 1+2, 2	72.98	50.26
1, 1+2, 2	2, 2+3, 3	71.98	48.55
1, 1+2, 2	3, 3+4, 4	73.78	51.63

表 28 输入图像尺寸 513×513 时本文模型的精度 %

第 1 组	第 2~4 组	mIoU	
		训练	验证
1, 1, 1+1, 1, 1	1, 1, 1+1, 1, 1	72.77	50.18
1, 1, 1+2, 2, 2	2, 2, 2+3, 3, 3	72.14	49.31
1, 1, 1+2, 2, 2	3, 3, 3+4, 4, 4	72.23	51.44
3, 3, 3+4, 4, 4	4, 4, 4+5, 5, 5	72.02	51.41

从表 28 可以看出,当使用 6 层相同扩张率叠加态时,与不使用空洞卷积的分支结构基准精度 50.18%相比,分支结构组合相同扩张率叠加态后,模型精度最高达到 51.44%,精度提升了 1.26%。

3.8 多尺度特征融合解码器分析

本组实验基于 DeepLabv3 模型将残差网络 ResNet-50 作为骨干网络,考虑训练过程中的计算量,修改 Block4 模块的空洞卷积扩张率为 1, 2, 4;在 ASPP 中添加 4 组分支 Block5 模块,第 1 组 Block5 模块空洞卷积扩张率为 1, 2, 4;再经过原模型的 $1 \times 1 \times 256$ 的普通卷积,第 2~4 组 Block5 模块空洞卷积扩张率为 2, 4, 8;之后分别经过原模型扩张率为 6, 12, 18 的 $3 \times 3 \times 256$ 卷积。

添加多尺度特征融合模块,通过导出不同 Block 模块输出的特征图像,实现不同尺度的特征图提取;采用逐像素相加的方式进行特征融合,再通过 3×3 ,步长为 1 的卷积输出,并经过双线性插值上采样将最终特征图还原到原始图像分辨率。

DeepLabv3 模型未添加/添加分支 Block5 模块的训练结果分别如表 29 和表 30 所示。

表 29 未添加分支 Block5 模块的精度 %

模型	mIoU	
	训练	验证
DeepLabv3 ^[14]	76.16	51.11
DeepLabv3 ^[14] +decoder-1	74.38	50.57
DeepLabv3 ^[14] +decoder-2	76.29	52.43
DeepLabv3 ^[14] +decoder-3	75.60	53.71
DeepLabv3 ^[14] +decoder-4	75.95	51.33
DeepLabv3 ^[14] +decoder-5	77.95	53.48
DeepLabv3 ^[14] +decoder-6	77.38	52.54

表 30 添加分支 Block5 模块的精度 %

模型	mIoU	
	训练	验证
DeepLabv3 ^[14] +Block5	75.38	51.29
DeepLabv3 ^[14] +Block5+decoder-1	74.61	50.69
DeepLabv3 ^[14] +Block5+decoder-2	75.72	52.72
DeepLabv3 ^[14] +Block5+decoder-3	75.78	53.71
DeepLabv3 ^[14] +Block5+decoder-4	73.91	50.78
DeepLabv3 ^[14] +Block5+decoder-5	76.32	53.19
DeepLabv3 ^[14] +Block5+decoder-6	77.42	56.04

通过表 29 可以看出, 添加多尺度特征融合解码器之后, 验证精度普遍提升. 观察逐层特征融合解码器 decoder-1~decoder-3 的验证精度发现, 深层特征的融合无法获得精度提升, 同时 decoder-1 出现精度下降的情况. 分析下降原因是深层特征通过空洞卷积获得, 像素的信息丢失没有得到有效的补充; 因为不同的深层特征缺失信息不存在相关性, 即使逐像素相加, 也无法获得较好的特征融合, 所以导致精度下降.

从表 30 可以看出, 在扩展分支机构后, 原始模型的精度略有提升, 从 51.11% 提升至 51.29%. 精度提升不高的原因是 Block5 模块的空洞卷积扩张率设置不合理, 可以通过调整扩张率的设置, 改善实验结果.

解码器 decoder-4~decoder-6 表示深度特征和一层浅层特征信息进行特征融合, 从表 29 和表 30 可以看出, 精度均有提升. 分析原因是浅层特征包含准确的定位信息, 使用浅层特征对缺失像素进行填充可增加采样像素之间的相关性.

扩展分支结构和添加多尺度特征融合解码器, 使模型的验证精度均有提升, 特别是融合最浅层特征的解码器 decoder-6, 验证精度达到 56.04%, 比 DeepLabv3 模型提升 4.75%.

3.9 其他模型

本文通过对比其他分割网络比较 BACNet 模型的效果, 实验结果如表 31 所示. 本文模型比主要的语义分割网络精度均有大幅提升, 与使用 Xception-65 骨干网络的 DeepLabv3+模型^[17]的精度 55.46% 相比, 使用 ResNet-50 为骨干网络的 BACNet 模型精度提高 0.96%; 使用 ResNet-101 为骨干网络的 BACNet 模型, 精度提高 1.77%.

为了验证 BACNet 模型的有效性, 将本文模型运用到不同的骨干网络, 分别使用 ResNet-50, ResNet-101 和 ResNet-152, 在验证精度和检测速度上进行定量比较, 实验结果如表 32 所示.

表 31 其他分割网络的精度对比 %

模型(骨干网络)	mIoU	
	训练	验证
VGG-16	56.87	37.60
VGG-19	56.56	38.47
FCN-16s(VGG-16) ^[6]	48.54	36.68
FCN-8s(VGG-16) ^[6]	18.59	19.61
DeepLabv2(VGG-16) ^[10]	44.76	35.11
DeepLabv2(VGG-19) ^[10]	37.00	30.52
DeepLabv2(ResNet-50) ^[10]	75.04	47.56
DeepLabv2(ResNet-101) ^[10]	54.77	32.88
DeepLabv3+(Xception-65) ^[17]		55.46
BACNet(ResNet-50)	74.71	56.42
BACNet(ResNet-101)	76.79	57.23
BACNet(ResNet-152)	73.73	55.55

表 32 其他骨干模型的验证精度和测试时间对比

模型(骨干网络)	验证 mIoU/%	测试时间/s
DeepLabv3(ResNet-50) ^[14]	51.28	0.093
DeepLabv3(ResNet-101) ^[14]	52.26	0.112
DeepLabv3(ResNet-152) ^[14]	52.13	0.125
BACNet(ResNet-50)	56.42	0.154
BACNet(ResNet-101)	57.23	0.177
BACNet(ResNet-152)	55.55	0.200

通过表 32 对比分析, 本文模型在不同的骨干网络中均达到了较高的验证精度, 在 ResNet-101 模型中验证精度最高, 达到 57.23%, 比 DeepLabv3 模型验证精度提升 4.97%. 使用 ResNet-50 骨干网络时本文模型比 DeepLabv3 模型提升了 5.14%; 使用 ResNet-152 骨干网络时本文模型比 DeepLabv3 模型提升了 3.42%.

对于机加工车间场景的语义分割测试结果如图 16 所示. 其中, DeepLabv3 模型的分割效果较差, 主要原因是使用空洞卷积时存在离散采样, 导致像素信息丢失和采样像素的相关性降低; 本文模型在 ResNet-50 骨干网络中, 对于道路线的分割更加精准, 随着骨干网络的深度增加, 分割图像目标类别的边缘更加精细.

4 结 语

本文提出一种分支空洞卷积神经网络模型 BACNet. 该模型扩展分支结构, 通过设置不同比例的空洞卷积扩张率实现调节特征图感受野, 以获取具有相关性的上下文信息; 通过空洞卷积的相

图 16 本文模型的测试结果

同扩张率叠加态,改善空洞卷积的棋盘效应,减少上下文信息的缺失问题;通过添加多尺度特征融合的解码器单元,利用目标定位相对准确的浅层特征和目标分类准确的深层特征进行特征融合,弥补由于棋盘效应导致的上下文信息缺失和像素信息不相关性的问题.实验结果表明,与使用 ResNet 骨干网络的 DeepLabv3 模型和使用 Xception-65 骨干网络的 DeepLabv3+模型相比, BACNet 具有更高的验证精度.

本文在提升模型精度的过程中扩展分支结构,增加多尺度特征融合解码器,在一定程度上导致模型的深度增加.一方面由于网络较深导致训练和测试时间相对更长,从表 32 可以看出, BACNet 比 DeepLabv3 模型在 ResNet-50 骨干网络中检测时间延长了 0.061 s;另一方面,由于增加了网络深度,导致模型的参数量明显增大,增加了模型的时空复杂度.因此,下一步的改进方案主要在于精简模型架构,在保持精度提升的情况下,缩减模型参数,降低时空复杂度.

参考文献(References):

- [1] Zhou Xiangming, Chen Yuanyi, Xu Huayin. Joint D*Lite algorithm and vision for AGV path planning and location[J]. Journal of Ordnance Equipment Engineering, 2020, 41(3): 126-130(in Chinese)
(周祥明, 陈员义, 徐华银. 联合 D*Lite 算法与视觉的 AGV 路径规划与定位[J]. 兵器装备工程学报, 2020, 41(3): 126-130)
- [2] Pang Y S, de la Cruz A L, Lodewijks G. Bipolar magnetic positioning system for automated guided vehicles[C] //Proceedings of the IEEE Intelligent Vehicles Symposium. Los Alamitos: IEEE Computer Society Press, 2008: 620-625
- [3] Xu Z X, Huang S L, Ding J C. A new positioning method for indoor laser navigation on under-determined condition[C] //Proceedings of the 6th International Conference on Instrumentation & Measurement, Computer, Communication and Control. Los Alamitos: IEEE Computer Society Press, 2016: 703-706
- [4] Osman K, Ghommam J, Saad M. Vision based lane reference detection and tracking control of an automated guided vehicle[C] //Proceedings of the 25th Mediterranean Conference on Control and Automation. Los Alamitos: IEEE Computer Society Press, 2017: 595-600
- [5] Zhang Jianpeng, Lou Peihuang, Qian Xiaoming, et al. Research on precise positioning technology by multi-window and real-time measurement for visual navigation AGV[J]. Chinese Journal of Scientific Instrument, 2016, 37(6): 1356-1363(in Chinese)
(张建鹏, 楼佩煌, 钱晓明, 等. 多窗口实时测距的视觉导引 AGV 精确定位技术研究[J]. 仪器仪表学报, 2016, 37(6): 1356-1363)
- [6] Shelhamer E, Long J, Darrell T. Fully convolutional networks

[1] Zhou Xiangming, Chen Yuanyi, Xu Huayin. Joint D*Lite algo-

- for semantic segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 640-651
- [7] Badrinarayanan V, Kendall A, Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495
- [8] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions[OL]. [2020-05-11]. <https://arxiv.org/abs/1511.07122>
- [9] Chen L C, Papandreou G, Kokkinos I, *et al.* Semantic image segmentation with deep convolutional nets and fully connected CRFs[OL]. [2020-05-11]. <https://arxiv.org/abs/1412.7062>
- [10] Chen L C, Papandreou G, Kokkinos I, *et al.* DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834-848
- [11] Lin G S, Milan A, Shen C H, *et al.* RefineNet: multi-path refinement networks for high-resolution semantic segmentation[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 5168-5177
- [12] He K M, Zhang X Y, Ren S Q, *et al.* Deep residual learning for image recognition[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778
- [13] Zhao H S, Shi J P, Qi X J, *et al.* Pyramid scene parsing network[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6230-6239
- [14] Chen L C, Papandreou G, Schroff F, *et al.* Rethinking atrous convolution for semantic image segmentation[OL]. [2020-05-11]. <https://arxiv.org/abs/1706.05587>
- [15] Deng J, Dong W, Socher R, *et al.* ImageNet: a large-scale hierarchical image database[C] //Proceedings of the IEEE Computer-Society Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2009: 248-255
- [16] Wang P Q, Chen P F, Yuan Y, *et al.* Understanding convolution for semantic segmentation[C] //Proceedings of the IEEE Winter Conference on Applications of Computer Vision. Los Alamitos: IEEE Computer Society Press, 2018: 1451-1460
- [17] Chen L C, Zhu Y K, Papandreou G, *et al.* Encoder-decoder with atrous separable convolution for semantic image segmentation[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 833-851