论一文一分享

报告人:郭晓龙



PointRend

ROS

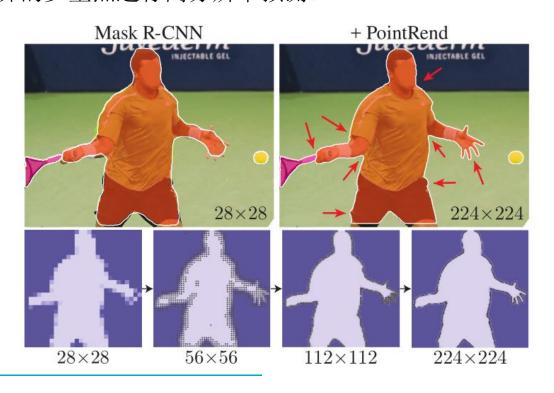
PointRend: Image Segmentation as Rendering 视图像分割为渲染 Facebook AI Research (FAIR) 2019年12月

Achieve Crisp object boundaries & Significant gains on COCO and Cityscapes & Unprecedented output resolution

链接: http://arxiv.org/abs/1912.08193

简介

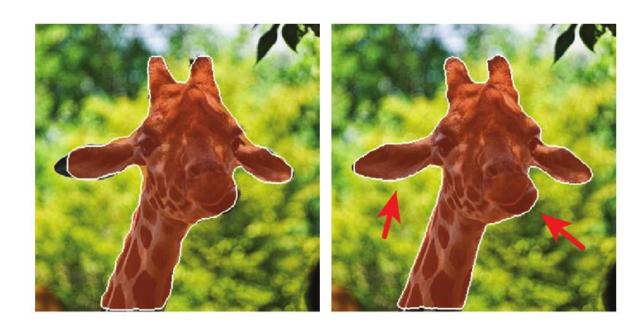
论文要解决的是图像分割质量问题,往往图像分割在物体边界处的分割质量很差,不能细致的分割出每个细节。因此作者提出了针对目标轮廓进行细化预测的一个模型: PointRend, 其思想是以迭代的方式细化从目标轮廓区域选择的点的分割预测,从而提升目标轮廓分割质量。提出的模型适用于实例分割和语义分割,能预测清晰的目标轮廓,同时也提升了相应的分割精度。如下图,每一步对平滑区域进行双线性上采样,对那些有可能是物体边界的少量点进行高分辨率预测。



动机

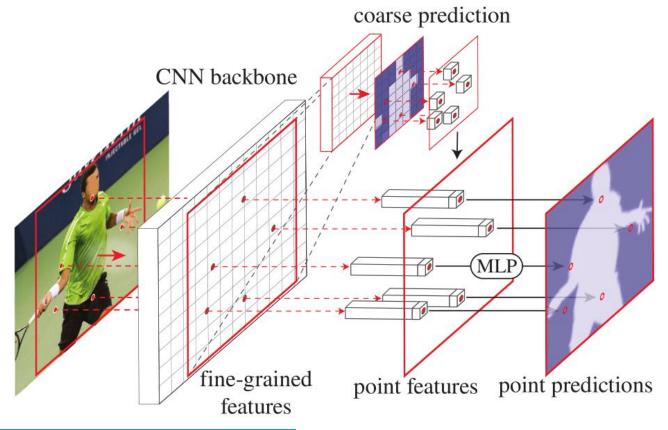
图像分割任务,通常使用CNN进行操作,输入是由像素组成的规则网格,输出是规则网格上的标签图(label map)。对于实例分割,标签图是一个二进制的前景掩码。

作者认为这种规则网格(box)不一定是理想的图像分割计算方式,在图像或特征这种规则的像素网格进行预测时,不可避免的存在平滑区域过采样,在物体轮廓欠采样。类比经典计算机图形学领域中同样存在的过采样和欠采样问题,将图像分割视为一个渲染问题,提出了PointRend。



Motivation

PointRend模块包含3个部分: (1)点选择策略,对少量选择的点预测其在高分辨率图中的类别; (2)提取对选择的每个点的特征表示; (3) point head,根据每个点的特征表示预测类别标签。

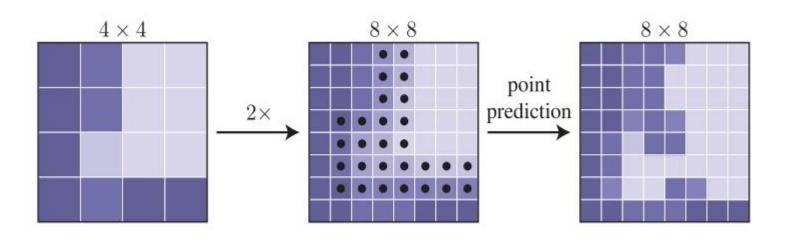


一个轻量级的分割头,为每个被检测对象(红色框)生成一个粗糙的(例如7×7)掩码。为了细化粗糙掩码,PointRend选择一组点(红点),用一个小的MLP分别对每个点进行分割。

PointRend Method

点选择

该方法的核心是灵活自适应地选择图像平面上的点预测分割标签。直观地说,这些点密集地位于高频区域附近,例如物体边界。在每次迭代中,PointRend使用双线性插值对之前预测的分割进行采样,然后在这个密度更大的网格上选择N个最不确定的点(例如,对于二进制掩码来说,概率接近0.5的点)。重复这个过程,直到分割结果被向上采样到所需的分辨率。



用双线性插值的方法对4×4网格的预测进行2×上采样。然后,PointRend对N个最模糊的点(黑点)进行预测,以恢复更精细网格上的细节。重复此过程,直到达到所需的网格分辨率。

Point Selection

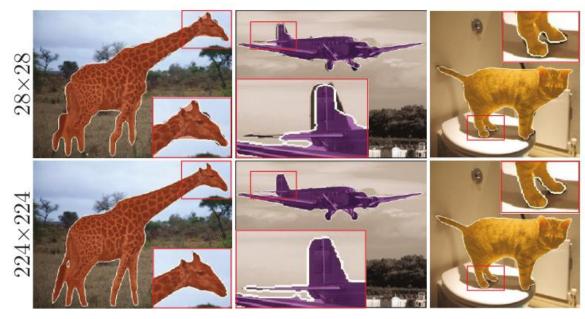
点向特征表示与Point head

PointRend在选定的点上,通过结合两种类型的特征构造点向特征:

细粒度特征:在每个采样点处从CNN feature maps上采用双线性插值提取256维特征向量。 提取出的细粒度特征是单个点的多通道特征,细粒度特征具有物体的细节信息。

粗预测特征: 粗预测特征对于实例分割来说是从Rol特征上插值提取K维特征向量。粗预测特征提供更多的上下文信息,同时表达语义类别。例如,在mask R-CNN中,粗预测可以是7×7分辨率的掩码输出。拼接两种类型的特征作为每个点的最终特征表示。

Point head基于这种特征表示,为每个独立点预测一个分割标签。

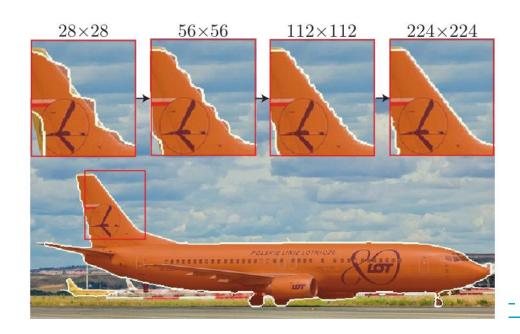


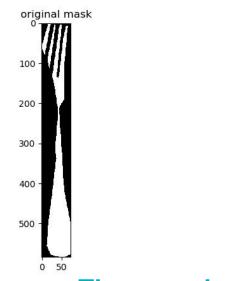
ronn-wise representation

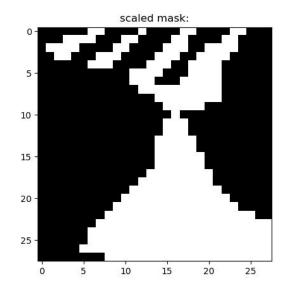
实验结果

		COCO		
mask head	backbone	AP	AP^{\star}	
4× conv	R50-FPN	37.2	38.1	
PointRend	R50-FPN	38.2 (+1.0)	40.1 (+2.0)	
4× conv	R101-FPN	38.6	40.2	
PointRend	R101-FPN	39.8 (+1.2)	42.0 (+1.8)	
4× conv	X101-FPN	39.5	40.7	
PointRend	X101-FPN	40.9 (+1.4)	43.4 (+2.7)	

	# points per	COCO		Cityscapes
output resolution	subdivision step	AP	AP^{\star}	AP
28×28	28 ²	36.1	37.7	35.4
56×56	28^{2}	36.2	38.2	<u>35.8</u>
112×112	28^{2}	36.3	38.3	35.8
224×224	28^{2}	36.3	38.3	35.8
224×224	14 ²	36.1	38.0	35.5
224×224	28^{2}	36.3	38.3	<u>35.8</u>
224×224	56 ²	36.3	38.3	35.8
224×224	112 ²	36.3	38.3	35.8

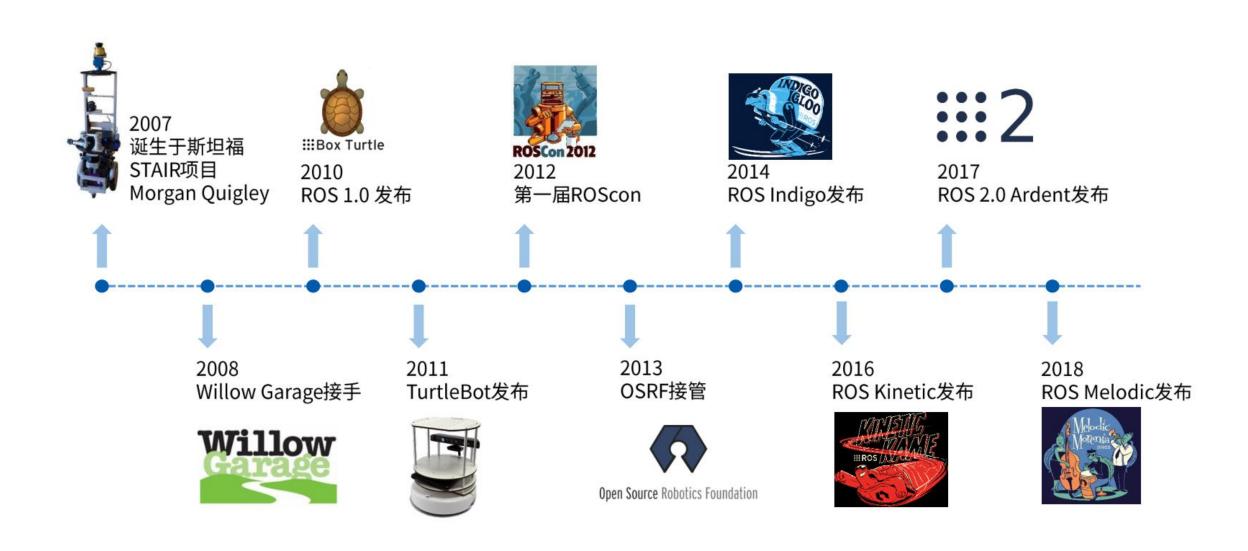






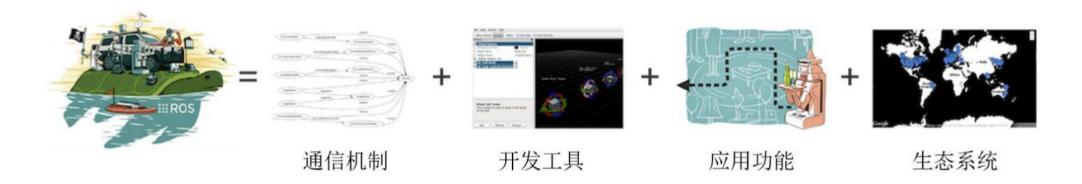
The experimental results

ROS发展史



ROS

ROS是什么



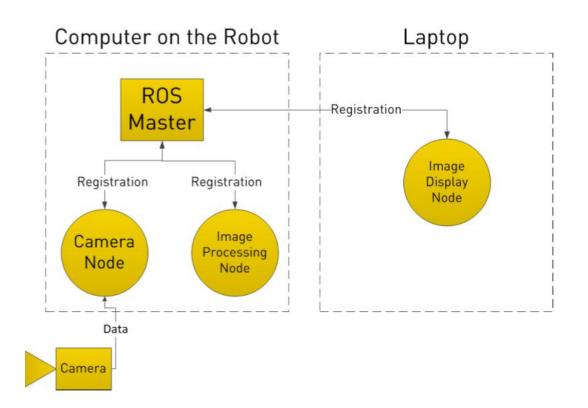


提高机器人研发中的软件复用率。

What is the ROS

ROS中的通信机制

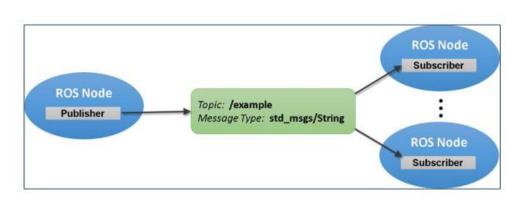
- **节点(Node)**——执行单元
 - 执行具体任务的进程、独立运行的可执行文件;
 - 不同节点可使用不同的编程语言,可分布式运行 在不同的主机;
 - 节点在系统中的名称必须是唯一的。
- **节点管理器 (ROS Master)** —— 控制中心
 - 为节点提供命名和注册服务;
 - 跟踪和记录话题/服务通信,辅助节点相互查找、 建立连接;
 - 提供参数服务器,节点使用此服务器存储和检索 运行时的参数。



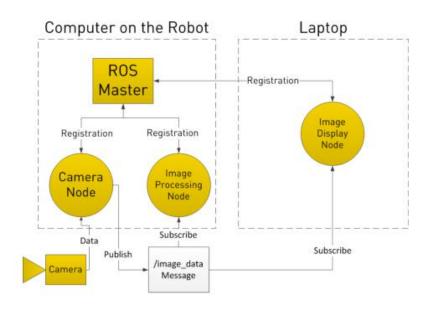
Communication mechanism

话题通信

- 话题(Topic)—— 异步通信机制
 - 节点间用来传输数据的重要总线;
 - 使用发布/订阅模型,数据由发布者传输到订阅者,同一个话题的订阅者或发布者可以不唯一。
- 消息 (Message) —— 话题数据
 - 具有一定的类型和数据结构,包括ROS提供的标准类型和用户自定义类型;
 - 使用编程语言无关的.msg文件定义,编译过程中生成对应的代码文件。



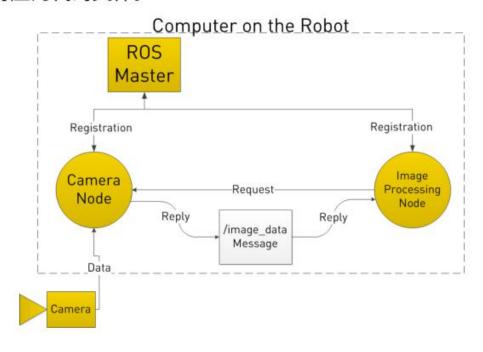
话题模型(发布/订阅)

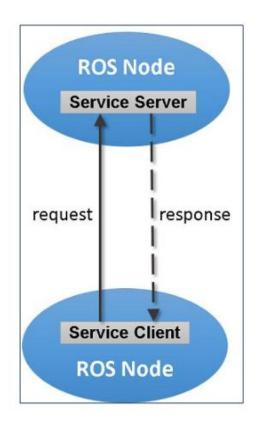


Topic communication

服务通信

- 服务 (Service) —— 同步通信机制
 - 使用客户端/服务器(C/S)模型,客户端发送请求数据,服务器完成 处理后返回应答数据;
 - 使用编程语言无关的.srv文件定义请求和应答数据结构,编译过程中 生成对应的代码文件。

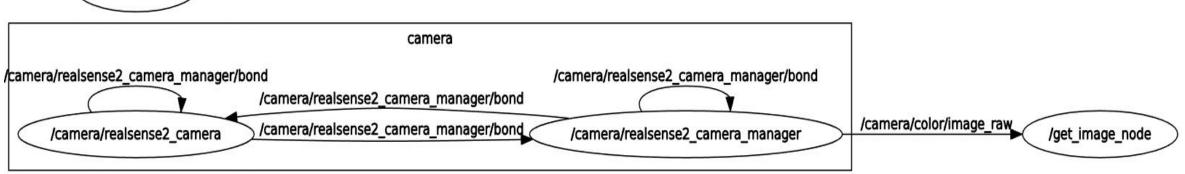


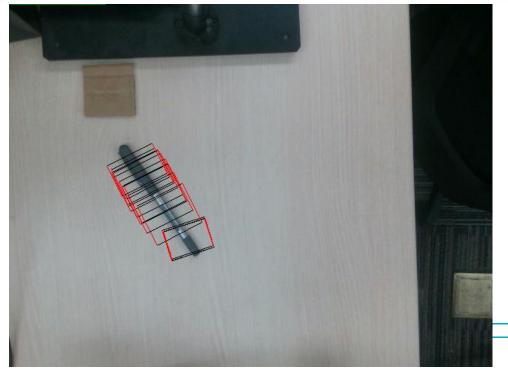


服务模型(请求/应答)

节点计算图

/grasp_client





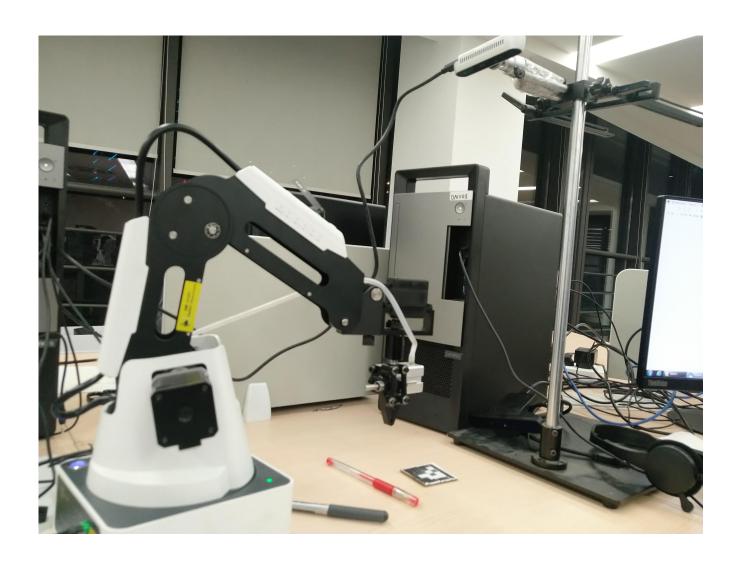
目前计算图中包含机械臂手节点,图像节点,抓取检测节点。

图像节点发布图像数据;

抓取检测节点订阅图像节点消息, 获取图像数据,并发布抓取框数据;

机械手节点订阅抓取检测节点的 消息,获取抓取框数据,执行抓取动 作。

Calculation chart



请您指导

谢谢~