

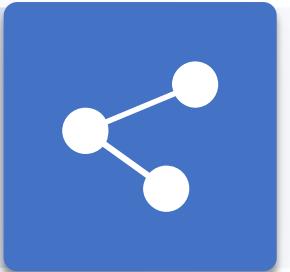


LiDAR R-CNN: An Efficient and Universal 3D Object Detector

Zhichao Li, Feng Wang, Naiyan Wang

2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

智能网络与优化实验室





1

介绍



介绍

对于自动驾驶车辆和机器人来说，在复杂的现实环境中估计周围物体的 7 个自由度（位置、尺寸和方向）状态是一项至关重要的任务。目前，基于激光雷达的 3D 目标检测由于其直接三维测量的能力而受到越来越多的关注。然而，与发达的 2D 图像检测相比，基于激光雷达的 3D 检测仍然存在点稀疏性和 3D 空间搜索空间大的困难。

激光雷达的点云是不规则的，大多数 3D 检测方法将点云数据转换为规则的 3D 体素网格或投影到 2D 视图，虽然这些方法可以通过使用 2D 或 3D 卷积提取特征，但体素或多视图特征构造中的量化误差限制了它们的性能。相反，基于点的方法可以从原始点云中学习特征，但通常需要复杂而低效的操作来提取局部信息。



文章贡献

文章提出了二阶段网络。其中第一阶段使用常见的体素化方法进行检测，排除掉大部分的背景点，之后在第一阶段的 proposal 的基础上，利用原始点云信息来得到更精确的边界框。由于 proposal 中的大部分点都是前景点，点云数量大大降低，所以第二阶段的网络可以完美规避基于点的方法计算量大的缺点，同时保留住了原始点云中目标的精确几何形状。

- 提出了一种基于 PointNet 的 R-CNN 型 3D 目标检测器。方法是对任何现有的 3D 探测器进行即插即用，不需要对基础探测器进行再训练。
- 揭示了使用基于点的 R-CNN 检测器时的尺寸模糊问题。通过仔细分析，我们提出了几种不同的方法来让检测器知道 proposal 框的大小。尽管设计简单，但它实现了显著的性能改进。
- 在 WOD 和 KITTI 数据集上测试了提出的方法，这些数据集带有各种基本检测器。



2

论文思路



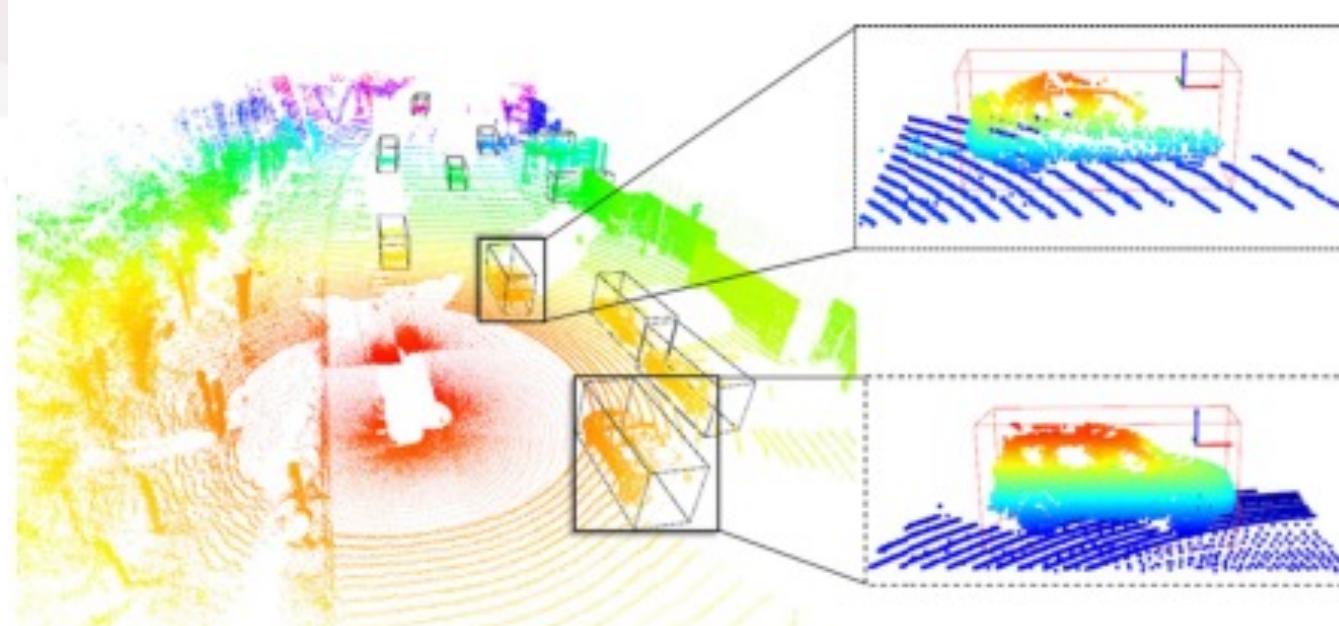
基于点的 R-CNN

1. 输入特征 (Input Features)

假设已经经过第一阶段网络，得到了点云中的 proposal。对于每一个3D proposal

$$b_i = (x_i, y_i, z_i, w_i, l_i, h_i, \theta_i)$$

可以扩大它的宽度和长度，以包含更多包含语义信息的点。然后将 proposal 中的所有点作为 R-CNN 的输入数据。为了提高 R-CNN 模型的泛化性，根据 3D 边界框坐标系将点对齐到 proposal 坐标系下：原点设置为 proposal 的中心，航向方向设置为 x 轴，其水平正交方向为 y 轴，垂直向上方向为 z 轴。

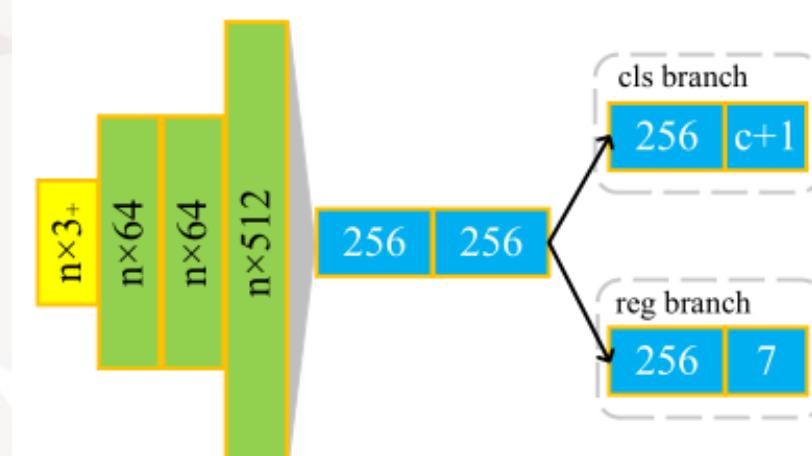




基于点的 R-CNN

2. 主体部分 (Backbone)

选择 PointNet 作为 backbone，它由一个具有 3 个全连接层的多层感知 (MLP) 模块和一个用于特征聚合的最大池操作符组成。然后将聚合后的特征分为两个分支：一个用于分类，另一个用于回归。



输入有 3 个或多个 channel , c 表示分类结果的类别数 , 而 $c + 1$ 表示背景类。



基于点的 R-CNN

3. 损失函数 (Loss Function)

对于分类分支，损失函数为

$$\mathcal{L}_{cls} = \frac{1}{B} \sum_{i=1}^B -\log p_{y(i)}^{(i)}$$

式中 B 表示 batch size , $y(i)$ 表示第 i 个样本的标签 , $p_{y(i)}^{(i)}$ 表示在 $y(i)$ 上的 softmax 概率。



基于点的 R-CNN

3. 损失函数 (Loss Function)

对于回归分支，由于前面将 proposal 框 $b_i = (x_i, y_i, z_i, w_i, l_i, h_i, \theta_i)$ 变为了 $\hat{b}_i = (0, 0, 0, w_i, l_i, h_i, 0)$ ，那么 3D ground truth 的框 $b^{gt} = (x_i^{gt}, y_i^{gt}, z_i^{gt}, w_i^{gt}, l_i^{gt}, h_i^{gt}, \theta_i^{gt})$ 也就变为

$$\hat{b}_i^{gt} = (x_i^{gt} - x_i, y_i^{gt} - y_i, z_i^{gt} - z_i, w_i^{gt}, l_i^{gt}, h_i^{gt}, \theta_i^{gt} - \theta_i)$$

按照 2D R-CNN，对中心点和框大小的回归定义为

$$t_i^c = \left(\frac{x_i^{gt} - x_i}{w_i}, \frac{y_i^{gt} - y_i}{l_i}, \frac{z_i^{gt} - z_i}{h_i} \right)$$

$$t_i^s = \left(\log \frac{w_i^{gt}}{w_i}, \log \frac{l_i^{gt}}{l_i}, \log \frac{h_i^{gt}}{h_i} \right)$$



基于点的 R-CNN

3. 损失函数 (Loss Function)

对于方向的回归则由于点云的稀疏性和某些对象的外观模糊性，它们的预测方向可能会翻转180度，因此预测目标应表示为

$$t_i^o = \begin{cases} \Delta\theta_i, & \Delta\theta_i \leq \frac{\pi}{2} \\ \Delta\theta_i - \pi, & \Delta\theta_i \geq \frac{\pi}{2} \end{cases}, \text{ where } \Delta\theta_i = (\theta_i^{gt} - \theta_i) \bmod \pi$$

回归目标为 $t_i = (t_i^c, t_i^s, t_i^o)$ ，损失函数描述为

$$\mathcal{L}_{reg} = \frac{1}{B_+} \sum_{i=1}^{B_+} SmoothL1(o_i - t_i), \text{ where } SmoothL1(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{other} \end{cases}$$

式中 B_+ 表示正样本数量， o_i 为回归分支的输出。最终可得损失函数为

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{reg}$$

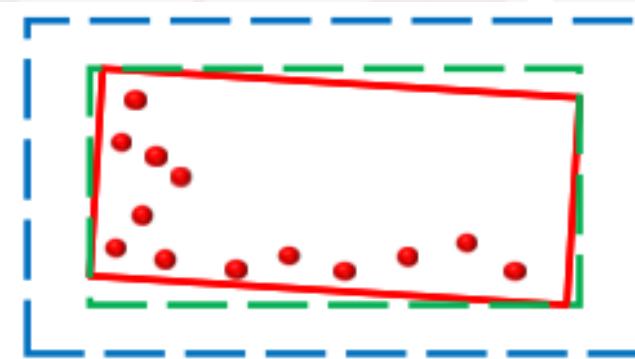
λ 为平衡因子，实验中可以取 20。



尺寸模糊问题

2D 图像中像素空间是规则且稠密的，但 3D 点云空间是非常稀疏且无规则的。如果直接使用 point-based 的方法，网络仅仅只会考虑 proposal 中存在的点，而 proposal 中的空白部分无法在特征中表示，如下图所示：如果只考虑其中的红色点，那么蓝色和绿色的边界框具有相同的特征表示，但是它们的分类和回归目标可能会差异很大（以红框为 ground truth）。分析原因发现：R-CNN 网络并不知道 proposal 的大小，那它如何能判断 proposal 是否准确、如何能在 proposal 的基础上回归出更准确的框呢？

我们将这个由 proposal 大小带来的问题称为尺寸歧义问题 (Size Ambiguity Problem)，那就需要把 proposal 的大小告诉 R-CNN。文章提出了 5 种解决方法。



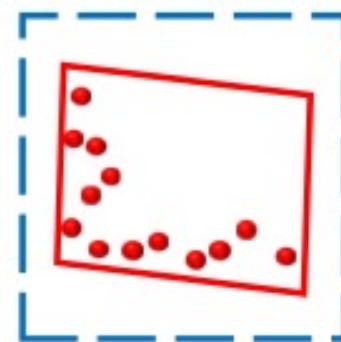
(a) Size ambiguity



尺寸模糊问题

1. 归一化 (Normalization)

最简单的解决方案是根据 proposal 的大小来归一化点的坐标。将 proposals 的边界对齐到 $(-0.5,0.5)$ 。如下图所示，归一化坐标会扭曲目标的形状，如果任务是单分类，这样做确实解决了问题，但在多分类情况下，虽然这样可以将 proposal 的大小隐式地告诉 R-CNN，但归一化之后物体的形状就改变了，忽略了不同类别尺寸的差异（一辆大卡车和一个行人压缩到同一个尺度，并不利于 PointNet 判断其类别），归一化使得模型难以区分目标类别。



(b) Size normalization

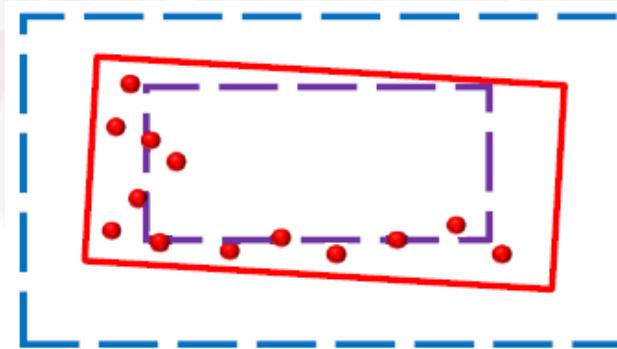


尺寸模糊问题

2. 锚框 (Anchor)

锚框是指将图片（点云）的采样框设置为固定大小，从而得到固定大小的采样结果。

为每个类别设置一个 anchor，这样回归目标会基于固定的 anchor，消除了尺度模糊问题。但是我们的目标是判断 proposal 质量的好坏并改进，而不是判断 anchor 的好坏。由于网络仍然不知道 proposal 的边界，该方法不能解决分类模糊性。此外，如果 box 中点很少（如下图紫框），不同种类目标特征可能相同，在这种情况下，回归会存在歧义。



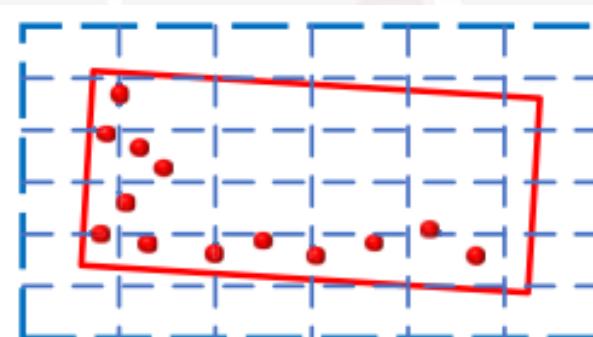
(c) Anchor



尺寸模糊问题

3. 体素化 (Voxelization)

参照 2D 图像的像素表示形式将 proposal 体素化，形成类似像素的规则方格。但 2D 图像上不存在这个歧义问题的主要原因是图像是非常致密的，不存在前景的地方会有背景的存在，只要网络能区分前景和背景，当 proposal 变大时，网络会发现背景变大了，从而给出 proposal 不准的信息。但 3D 点云中，proposal 变大时，网络虽然能看到最外围的 voxel 中没有点的存在，但网络所能感知到的边界只能到达 voxel 级别，而无法做到 point 级别的精度。而且如果对于每个类别都设置同样数量的 voxel，也会存在类似第一种尺寸归一化带来的多类别分类不容易区分的问题。



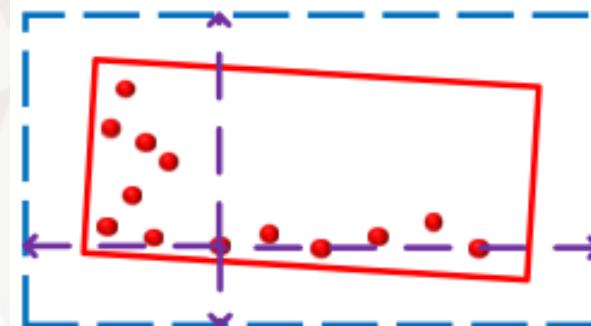
(d) Voxelization



尺寸模糊问题

4. 边界距离 (Boundary Offset)

既然原因是网络不知道边界的位置，最简单的方法是将 proposal 的边界信息作为输入告诉网络，在每个点后面串联上它们到上下左右前后 6 条边的归一化距离。从偏移量出发，网络将能够知道这些点离 proposal 的边界有多远，这就可以解决尺寸模糊问题。



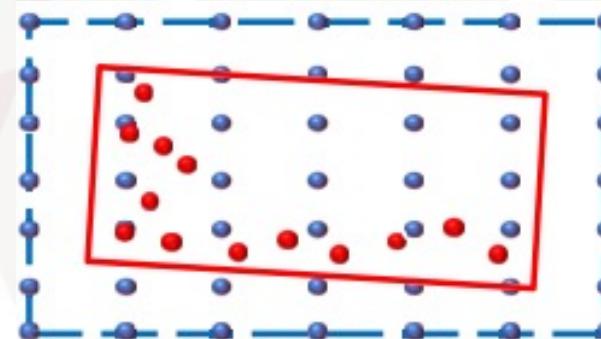
(e) Boundary offset



尺寸模糊问题

5. 虚点 (Virtual Points)

2D 图像上不存在歧义问题是因为图像上有背景的存在，而 3D 空间中目标的周围都是空的，无法通过背景点的多少来判断框的大小。从这个角度出发，如果将这片空间手动填满虚点，让这些虚点起到图像中背景的作用，应该也可以让 R-CNN 网络感知到 proposal 框的大小。具体做法是在 proposal 中均匀地放入网格状的虚点，这些虚点和真实的点通过额外加入的一维的 0-1 特征加以区分。这样网络通过最近的实点到边界上的虚点的距离，即可判断 proposal 框的大小。



(f) Virtual point



3

实验结果



实验结果

实验是在目前最大的公开自动驾驶数据集 Waymo Open Dataset (WOD) 上进行的，它包含约 16 万帧点云，超过 1200 万个目标 3D 框，远大于之前的公开数据集。选用常见的 SECOND 和 PointPillars 作为 baseline，直接将二者的输出（即 proposals）作为我们网络新的输入：直接从原始点云上抠取 proposal 内的点云 xyz，并将其旋转平移至 proposal 坐标系，之后将它们输入进 PointNet，由分类和回归两个 loss 来监督网络的训练。

在这个强 baseline 的基础上，套用第二阶段网络，单独对 Vehicle 进行 refine，结果如下：

其中 AP 表示 Average Precision，BEV 表示（鸟瞰图）

可以看到原始的 PointNet R-CNN 即可将 AP 提升 2.5 个点

而在解决了尺寸歧义问题后，

还能进一步提升 1.5 个点，达到 75.6。

Methods	3D AP@70	BEV AP@70
PointPillars [12]	71.6	87.1
PointNet refinement	74.1	87.9
voxel	72.9	87.2
anchor	75.2	88.2
size normalization	75.4	88.1
virtual point	75.4	88.1
boundary offset	75.6	88.3



实验结果

然而在多分类场景下，原始的 PointNet R-CNN 就只能将 Vehicle 提升 0.5 个点 (71.6 到 72.1)，而在 Pedestrian (70.6 到 69.2) 和 Cyclist (64.4 到 62.2) 两个类别上甚至还会掉点，且一系列之前分析的几个方案 (voxel、anchor、normalization 等)，均存在或多或少的掉点现象，而本文提出的两种有效的策略均能实现显著涨点。

Methods	vehicle	pedestrian	cyclist
PointPillars [12]	71.6	70.6	64.4
PointNet refinement	72.1	69.2	62.2
voxel	72.1	69.8	64.5
anchor	72.5	70.2	63.5
size normalization	72.7	69.9	64.4
virtual point	73.3	70.4	66.2
boundary offset	73.4	70.6	66.8



实验结果

在此基础上，我们还做了进一步的实验，套用两次 Lidar R-CNN，将结果进一步提升：

LEVEL_1 和 LEVEL_2 表示 WOD 数据集两种不同的 ground truth 标定。

Difficulty	Method	vehicle (IoU=0.7)		pedestrian (IoU=0.5)		cyclist (IoU=0.5)	
		3D AP	3D APH	3D AP	3D APH	3D AP	3D APH
LEVEL_1	SECOND [44]	58.5	57.9	63.9	54.9	48.6	47.6
	LiDAR R-CNN (sec)	62.6	62.1	68.2	59.5	52.8	51.6
	PointPillars [12]	71.6	71.0	70.6	56.7	64.4	62.3
	LiDAR R-CNN (pp)	73.4	72.9	70.6	57.8	66.8	64.8
LEVEL_2	LiDAR R-CNN (2x)	73.5	73.0	71.2	58.7	68.6	66.9
	SECOND [44]	51.6	51.1	56.0	48.0	46.8	45.8
	LiDAR R-CNN (sec)	54.5	54.0	59.3	51.7	50.9	49.7
	PointPillars [12]	63.1	62.5	62.9	50.2	61.9	59.9
	LiDAR R-CNN (pp)	64.6	64.1	62.5	50.9	64.3	62.4
	LiDAR R-CNN (2x)	64.7	64.2	63.1	51.7	66.1	64.4



实验结果

与体素化的二阶段方法相比，网络无需抽取卷积特征，其输入就是原始的点云坐标，而且主干网络是一个很小的 PointNet，所以本文方法在速度上也比之前的二阶段方法要快很多，仅需 4.5ms 即可完成一帧点云的 refine：

Methods	3D AP	time	Params
voxel	72.9	19ms	3.5M
ours	75.6	4.5ms	0.5M



4

结论



结论

本文介绍了 LiDAR R-CNN，一种快速、准确的二级 3D 目标检测器。通过对尺度模糊问题的详细分析和深思熟虑的实验，文章提出了切实可行的解决方案。在 Waymo Open Dataset 上的综合实验表明，文章的方法可以在所使用的 baseline 模型上稳步提高。

多传感器融合感知系统是机器人技术和自动驾驶所必需的。除了在单帧激光雷达检测方面令人鼓舞的性能外，本文的激光雷达 R-CNN 很容易推广到其他类型的输入，例如多帧激光雷达和RGB+激光雷达。作为第二阶段框架，文章的方法更适合处理各种聚合输入。

源码

- https://github.com/TuSimple/LiDAR_RCNN



謝謝

Thank You

THANKS

