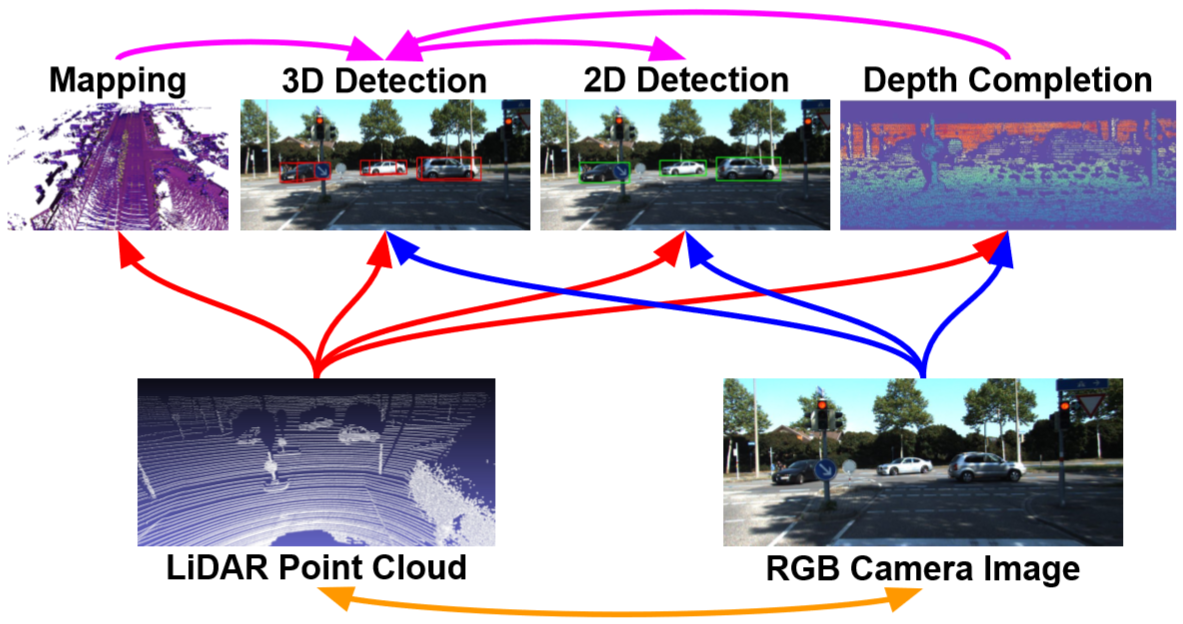
Fast R-CNN

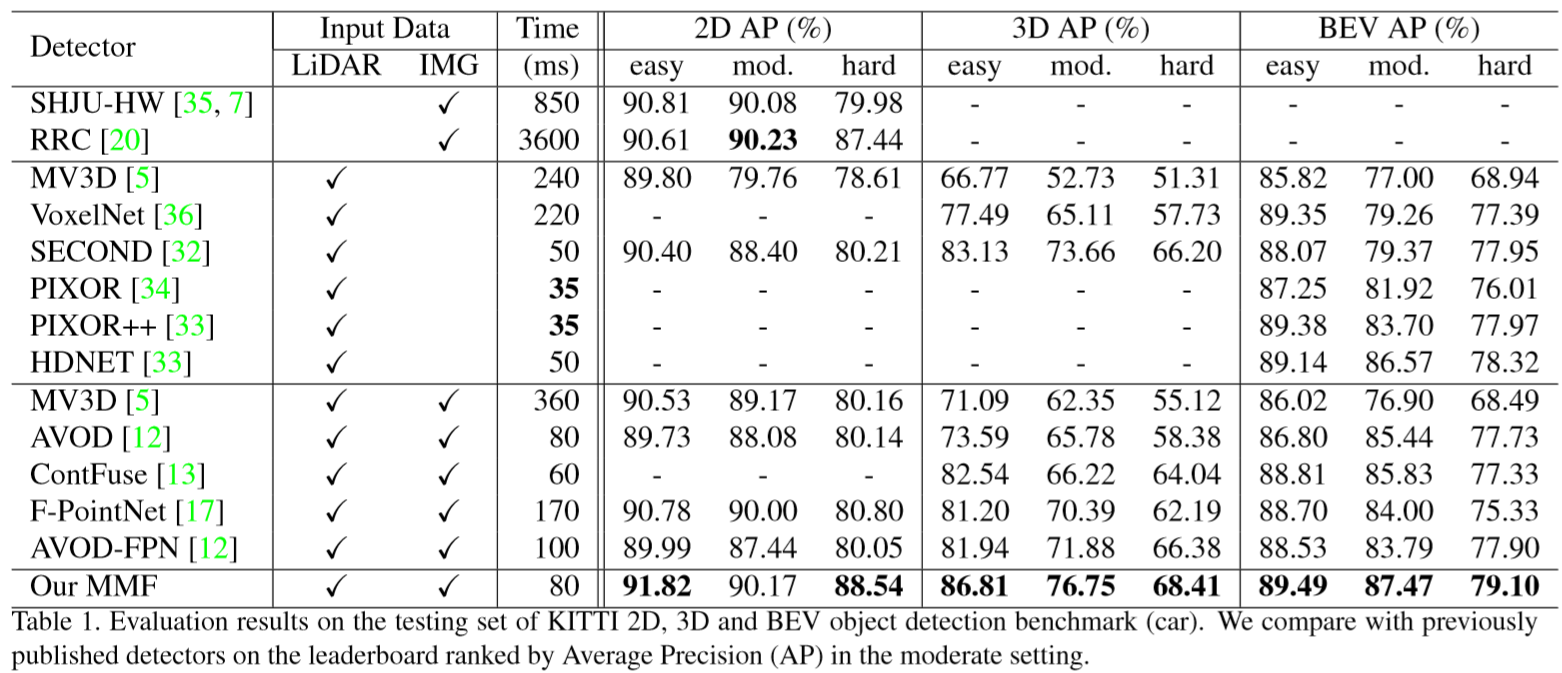
论文引用：

Girshick R. Fast r-cnn[C]//Proceedings of the IEEE International Conference on Computer Vision(ICCV). 2015: 1440-1448.

|  |  |
| --- | --- |
| ABSTRACT  This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9x faster than R-CNN, is 213x faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3x faster, tests 10x faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is available under the open-source MIT License at https://github.com/rbgirshick/fast-rcnn. | 摘要  本文提出了一种快速的基于区域的卷积网络方法（fast R-CNN）用于目标检测。Fast R-CNN建立在以前使用的深卷积网络有效地分类目标的成果上。相比于之前的成果，Fast R-CNN采用了多项创新提高训练和测试速度来提高检测精度。Fast R-CNN训练非常深的VGG16网络比R-CNN快9倍，测试时间快213倍，并在PASCAL VOC上得到更高的精度。与SPPnet相比，Fast R-CNN训练VGG16网络比他快3倍，测试速度快10倍，并且更准确。Fast R-CNN的Python和C ++（使用Caffe）实现以MIT开源许可证发布在：https://github.com/rbgirshick/fast-rcnn。 |
| 1. Introduction   Self-driving vehicles have the potential to improve safety, reduce pollution, and provide mobility solutions for otherwise underserved sectors of the population. Fundamental to its core is the ability to perceive the scene in real-time. Most autonomous driving systems rely on 3dimensional perception, as it enables interpretable motion planning in bird's eye view. | 1. 引文   自动驾驶车辆有可能提高安全性，减少污染，并为人口中服务不足的部门提供出行解决方案。其核心的基础是能够实时感知场景。大多数自动驾驶系统依赖于三维感知，因为它可以在鸟瞰视图中实现可解释的运动规划。 |



|  |  |
| --- | --- |
| *Figure 1. Differen*t sensors (bottom) and tasks (top) are complementary to each other. We propose a joint model that reasons on two sensors and four *tasks, and show that the target task - 3D object detection can beneﬁt from multi-task learning and multi-sensor fusion.* | *图1.不同的传*感器（底部）和任务（顶部）互为补充。我们提出了一个联合模型，它推导出两个传感器和四个任务，并表明目标任务 - *三维物体检测可以从多任务学习和多传感器融合中获益。* |
| We propose a new multi-sensor fusion architecture that leverages the advantages from both point-wise and ROI-wise feature fusion, resulting in fully fused feature representations. Knowledge about the location of the ground can provide useful cues for 3D object detection in the context of self-driving vehicles, as the trafﬁc participants of interest are restrained to this plane. Our detector estimates an accurate voxel-wise ground location online as one of its auxiliary tasks. This in turn is used by the bird's eye view (BEV) backbone network to reason about relative location. We also exploit the task of depth completion to learn better cross-modality feature representation and more importantly, to achieve dense point-wise feature fusion with pseudo LiDAR points from dense depth. | 我们提出了一种新的多传感器融合架构，它利用了逐点和逐ROI特征融合的优势，从而实现了完全融合的特征表示。关于地面位置的知识可以在自动驾驶车辆的背景下为3D物体检测提供有用的提示，因为感兴趣的交通参与者被限制在该平面上。我们的探测器在线估计精确的体素地面位置，作为其辅助任务之一。这反过来被鸟瞰（BEV）骨干网用于推断相对位置。我们还利用深度补全的任务来学习更好的跨模态特征表示，更重要的是，利用密集深度的伪LiDAR点实现密集的逐点特征融合。 |
| 1. Related Work   We review related works that exploit multi-sensor fusion and multi-task learning to improve 3D object detection.  **3D detection from single modality:** Early approaches to 3D object detection focus on camera based solutions with monocular or stereo images [3, 2]. However, they suffer from the inherent difﬁculties of estimating depth from images and as a result perform poorly in 3D localization. More recent 3D object detectors rely on depth sensors such as LiDAR [34, 36]. However, although range sensors provide precise depth measurements, the observations are usually sparse (particularly at long range) and lack the information richness of images. It is thus difﬁcult to distinguish classes such as pedestrian and cyclist with LiDAR-only detectors. | 1. **相关工作**   我们回顾了利用多传感器融合和多任务学习来改进3D对象检测的相关工作。  **单模态的3D检测：**早期的3D物体检测方法主要集中在基于摄像头的单目或立体图像解决方案[3,2]。然而，它们遭受了图像估计深度的固有困难，因此在3D定位中表现不佳。更新近的3D物体探测器依赖于深度传感器，例如LiDAR [34,36]。然而，尽管距离传感器提供精确的深度测量，但观测通常是稀疏的（特别是在远距离）并且缺乏图像的信息丰富度。因此，仅限LiDAR的探测器难以将诸如行人和骑车人的类别区分开来。 |
| 1. **Multi-Task Multi-Sensor Detector**   One of the fundamental tasks in autonomous driving is to perceive the scene in real-time. In this paper we propose a multi-task multi-sensor fusion model for the task of 3D object detection. We refer the reader to Figure 2 for an illustration of the model architecture. Our approach has the following highlights. First, we design a multi-sensor architecture that combines point-wise and ROI-wise feature fusion. Second, our integrated ground estimation module reasons about the geometry of the road. Third, we exploit the task of depth completion to learn better multi-sensor features and achieve dense point-wise feature fusion. As a result, the whole model can be learned end-to-end by exploiting a multi-task loss.  In the following, we ﬁrst introduce the architecture of the multi-sensor 2D and 3D detector with point-wise and ROI-wise feature fusion. We then show how we exploit the other two auxiliary tasks to further improve 3D detection. Finally we provide details of how to train our model end-to-end. | 1. **多任务多传感器探测器**   自动驾驶的基本任务之一是实时感知场景。在本文中，我们提出了一个多任务多传感器融合模型，用于三维物体检测任务。我们将读者引用到图2中以获得模型体系结构的图示。我们的方法有以下亮点。首先，我们设计了一种多传感器架构，它结合了逐点和逐ROI特征融合。其次，我们的综合地面估算模块有关道路几何形状的推理。第三，我们利用深度补全的任务来学习更好的多传感器特征并实现密集的逐点特征融合。因此，可以通过利用多任务损失来端到端地学习整个模型。  在下文中，我们首先介绍了具有逐点和逐ROI特征融合的多传感器2D和3D探测器的架构。然后，我们将展示如何利用其他两个辅助任务来进一步改进3D检测。最后，我们提供了如何对端到端训练模型的详细信息。 |
| 3.1 Fully Fused Multi-Sensor Detector  Our multi-sensor detector takes a LiDAR point cloud and an RGB image as input. The backbone network adopts the two-stream structure, where one stream extracts image feature maps, and the other extracts LiDAR BEV feature maps. Point-wise feature fusion is applied to fuse multiscale image features to BEV stream. The ﬁnal BEV feature map predicts dense 3D detections per BEV voxel via 2D convolution. After Non-Maximum Suppression (NMS) and score thresholding, we get a small number of high-quality 3D detections and their projected 2D detections (typically fewer than 20 when tested on KITTI dataset). We then apply a 2D and 3D box reﬁnement by ROI-wise feature fusion, where we combine features from both image ROIs and BEV oriented ROIs. After the reﬁnement, the detector outputs accurate 2D and 3D detections. | 3.1 多传感器完全融合探测器  我们的多传感器探测器采用LiDAR点云和RGB图像作为输入。骨干网采用双流结构，其中一个流提取图像特征映射，另一个流提取LiDAR BEV特征映射。逐点特征融合用于将多尺度图像特征融合到BEV流。最终的BEV特征图通过2D卷积预测每个BEV体素的密集3D检测。在非最大抑制（NMS）和得分阈值之后，我们获得少量高质量3D检测及其投影的2D检测（在KITTI数据集上测试时通常少于20）。然后，我们通过逐ROI特征融合应用2D和3D盒子改进，其中我们结合了来自图像ROI和BEV导向ROI的特征。完成后，探测器输出准确的2D和3D检测。 |



|  |  |
| --- | --- |
| 3.1.1 Feature Learning Network  **Voxel Partition:** Given a point cloud, we subdivide the 3D space into equally spaced voxels as shown in Figure 2. Suppose the point cloud encompasses 3D space with range D,H,W along the Z, Y, X axes respectively. We define each voxel of size vD,vH, and vW accordingly. The resulting 3D voxel grid is of size D′=D/vD, H′=H/vH, W′=W/vW. Here, for simplicity, we assume D,H,W are a multiple of vD,vH,vW.  **Grouping：**We group the points according to the voxel they reside in. Due to factors such as distance, occlusion, object‘s relative pose, and non-uniform sampling, the LiDAR point cloud is sparse and has highly variable point density throughout the space. Therefore, after grouping, a voxel will contain a variable number of points. An illustration is shown in Figure 2, where Voxel-1 has significantly more points than Voxel-2 and Voxel-4, while Voxel-3 contains no point.  **Random Sampling:** Typically a high-definition LiDAR point cloud is composed of ~100k points. Directly processing all the points not only imposes increased mem-ory/efficiency burdens on the computing platform, but also highly variable point density throughout the space might bias the detection. To this end, we randomly sample a fixed number, T of points from those voxels containing more than T points. This sampling strategy has two purposes, (1) computational savings (see Section 2.3 for details); and (2) decreases the imbalance of points between the voxels which reduces the sampling bias, and adds more variation to training.  **Stacked Voxel Feature Encoding:** The key innovation is the chain of VFE layers. For simplicity, Figure 2 illustrates the hierarchical feature encoding process for one voxel. Without loss of generality, we use VFE Layer-1 to describe the details in the following paragraph. Figure 3 shows the architecture for VFE Layer-1. | 3.1.1 特征学习网络  **体素分区：**给定点云，我们将3D空间细分为等间距的体素，如图2所示。假设点云包含分别沿Z，Y，X轴的范围D,H,W的3D空间。我们相应地定义大小为vD，vH和vW的每个体素。得到的3D体素网格的大小为D′=D/vD, H′=H/vH, W′=W/vW。这里，为简单起见，我们假设D,H,W是vD,vH,VW的倍数。  **分组：**我们根据它们所处的体素对点进行分组。由于距离，遮挡，物体的相对姿势和非均匀采样等因素，LiDAR点云稀疏并且在整个空间中具有高度可变的点密度。因此，在分组之后，体素将包含可变数量的点。图2中示出了图示，其中Voxel-1具有比Voxel-2和Voxel-4多得多的点，而Voxel-3不包含点。  **随机采样：**通常，高清晰度LiDAR点云由~100k点组成。直接处理所有点不仅会增加计算平台上的内存/效率负担，而且整个空间中高度可变的点密度可能会使检测偏差。为此，我们从包含超过T点的那些体素中随机采样固定数量的T点。这种抽样策略有两个目的，(1)计算节省（详见2.3节）; (2)减少体素之间点的不平衡，减少采样偏差，并增加训练的变化。  **堆叠体素特征编码:** 关键创新是VFE层链。为简单起见，图2说明了一个体素的分层特征编码过程。在不失一般性的情况下，我们使用VFE Layer-1来描述以下段落中的细节。图3显示了VFE Layer-1的体系结构。 |

1. References

[1] Beltrán J, Guindel C, Moreno F M, et al. Birdnet: a 3D object detection framework from LiDAR information[C]//2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018: 3517-3523.

[2] Chen X, Kundu K, Zhang Z, et al. Monocular 3d object detection for autonomous driving[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016: 2147-2156.

[3] Chen X, Kundu K, Zhu Y, et al. 3d object proposals for accurate object class detection[C]//Advances in Neural Information Processing Systems (NIPS). 2015: 424-432.

[4] Chen X, Kundu K, Zhu Y, et al. 3d object proposals using stereo imagery for accurate object class detection[J]. IEEE transactions on pattern analysis and machine intelligence (PAMI), 2017, 40(5): 1259-1272.

[5] Chen X, Ma H, Wan J, et al. Multi-view 3d object detection network for autonomous driving[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017: 1907-1915.