

## 4D 毫米波雷达和单目摄像头视觉融合算法

### 一、赛题背景

传统的 3D 毫米波雷达点云因点云极其稀疏且缺少关键的高度信息通常用作目标检测的辅助信息。新一代的高分辨率 4D（3D 空间位置+1D 速度）毫米波雷达具有更高的分辨率和俯仰角高度测量能力，有着出色的 3D 目标检测潜力。但由于其进入市场时间较晚，目前 4D 毫米波雷达点云与视觉融合的目标检测算法相关研究还比较少，4D 雷达数据集较少，且都是在长距离下采集的数据集。因此本赛题提供道路环境下车载 4D 毫米波雷达和摄像头采集数据，要求参赛团队能够基于该数据集完成雷视融合 3D 目标检测。主要考核参赛团队对数据增强、雷视数据对齐、特征融合等关键技术的掌握和创新能力。

### 二、赛题应用场景

在智能驾驶和自动驾驶技术快速发展的背景下，多传感器融合技术成为提升车辆环境感知能力的关键。4D 毫米波雷达和单目摄像头作为两种互补性极强的传感器，分别在不同条件下表现出色。毫米波雷达能够在恶劣天气（如雨、雾、雪）和光照不足的环境中提供可靠的检测，而单目摄像头则在良好天气和光照条件下提供高精度的图像信息。然而，单一传感器在某些情况下存在局限性，例如毫米波雷达在分辨率和细节捕捉上相对较弱，而摄像头在远距离检测和复杂环境中的表现不足。借助视觉融合算法和人工智能技术在复杂多变的环境中实现对车辆、行人、障碍物等目标的准确检测、分类和跟踪，提升自动驾驶系统的感知能力和安全性。

### 三、赛题任务

参赛者需利用主办方提供的高分辨率 4D 毫米波成像雷达密集点云数据集，设计并实现人工智能算法，对道路目标检测进行自动检测和分类。具体任务包括：

**1. 3D 目标检测：**精准定位目标的 3D 位置和边界框。

**2. 分类：**判断目标属于哪一个分类。

### 四、数据集及数据说明

#### （一）数据来源

数据是在城市道路上自行采集的，数据采集平台如图 4.1 所示，高分辨率 4D 毫米波雷达 OCULII-EAGLE 和摄像头 Intel RealSense D435i 安装在车辆顶部的中心位置，距离地面约 1.85 米。为了确保 4D 雷达点云的密度并生成密集雷达点云，我们在数据采集过程中选择了短距离模式。在此模式下，如图 4.2 所示高分辨率 4D 毫米波雷达 OCULII-EAGLE 的探测距离为 0m-25m，方位角范围为  $-56.5^{\circ} \sim 56.5^{\circ}$ ，俯仰角范围为  $-2.5^{\circ} \sim 22.5^{\circ}$ ，每秒采集 12 帧点云。



图 4.1 数据采集平台

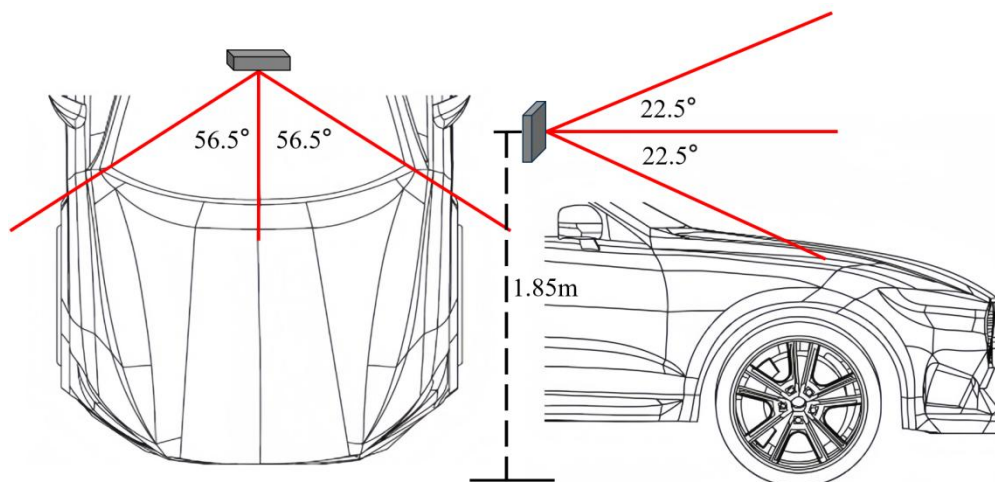


图 4.2 高分辨率 4D 毫米波雷达可视范围示意图

采集到的 4D 毫米波雷达点云数据格式为[距离，方位角，俯仰角，多普勒，功率， $x, y, z$ ]。其中“功率（dB）”为探测信号的信噪比；“多普勒”为目标与雷达之间的相对速度；“距离，方位角，俯仰角”为极坐标；“ $x, y, z$ ”为根据极坐标计算的笛卡尔坐标。Intel RealSense D435i 摄像头的视场角 FOV 为  $69^{\circ} \times 42^{\circ}$ （H×V），分辨率为  $640 \times 480$ ，帧率为每秒 30 帧图像。

## （二）参数标定

传感器参数标定是确保自动驾驶系统多传感器协同工作的基本要求。良好的标定保证了不同传感器采集的数据能够在同一坐标系下准确对齐，为感知系统提供了一致

的认知基础。这包括确定传感器内部映射关系（内参标定），如相机焦距和畸变参数，以及确定传感器与其他坐标系之间的转换关系（外参标定），例如相机和 4D 毫米波雷达之间的相对位置和方向。标定的准确性直接影响了自动驾驶系统对周围环境的理解和决策的可靠性。

为确保后续雷达与点云融合工作的稳定与精确，需要对 4D 毫米波雷达与相机进行参数标定。如图 4.1 数据采集平台所示，4D 毫米波雷达与图像采用不同的坐标系。本题以 4D 毫米波雷达坐标系作为世界坐标系，车辆前进方向为 X 轴正向，Y 轴正向为前进方向左向，Z 轴正向为竖直向上。

将 4D 毫米波雷达点云所在的世界坐标系与相机图像所在的像素坐标系联系起来需要坐标系的三次变换。在不考虑图像畸变的情况下，首先需要世界坐标系到相机坐标系的刚体变换，然后是相机坐标系到图像坐标系的透视投影，最后是从图像坐标系到像素坐标系的仿射变换。

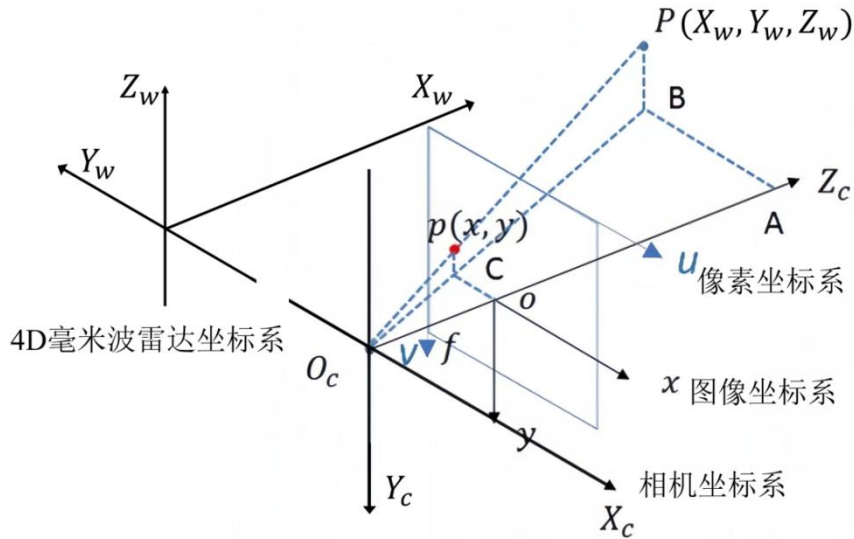


图 4.3 多传感器坐标系关系模型

如图 4.3 所示，如果想要将 4D 毫米波雷达点云所在的世界坐标系下的点  $P_w(X_w, Y_w, Z_w)$  与图像中像素坐标  $p(u, v)$  关联起来，首先需要使用公式 (4.1) 将点通过旋转和平移操作转换到相机坐标系中

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

(4.1)

其中， $R$  为旋转矩阵， $T$  为平移矩阵。本文中我们通过对齐安装的方式使相机与 4D 毫米波雷达朝向一致向前、距离相近、高度相同。并通过手工测量的方法得到两个传感器之间距离参数，并通过多帧图像中关键目标点云投影与图像物体对比的方式对参数进行手工微调，最终得到标定后的外参矩阵如下：

$$\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & -0.25 \\ 0 & 0 & -1 & 0.4 \\ 1 & 0 & 0 & -0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

得到相机坐标系下的点  $P_c(X_c, Y_c, Z_c)$  后可以通过透视投影并将原点平移到左上角获得像素坐标系下的像素点  $p(u, v)$ ，矩阵形式的公式如下：

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = KP_c \quad (4.2)$$

其中， $c_x, c_y$  为原点平移的像素距离， $f_x = f/d_x, f_y = f/d_y$ ， $f$  为焦距， $d_x, d_y$  为单个像素的物理宽度， $K$  表示内参矩阵。我们使用相机出厂时得到的标准标定的内参矩阵作为转换内参矩阵：

$$K = \begin{bmatrix} 605.6403 & 0.0 & 319.2964 & 0.0 \\ 0.0 & 605.6746 & 235.4414 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix}$$

综合公式 (4.1) 与 (4.2) 可得，4D 毫米波雷达点云所在的世界坐标系与相机图像所在的像素坐标的转换关系如下：

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ T_w \\ Z_w \\ 1 \end{bmatrix} \quad (4.3)$$

由此我们建立 4D 毫米波雷达与图像之间的关系矩阵。通过此公式可以将点云投



影至图像当中。

### （三）数据规模

初赛共提供 6551 个样本帧，其中训练集包含 5169 个样本帧，用于参赛者训练算法模型；验证集由参赛选手自主划分，用于模型调优与性能评估；测试集包含 1382 个样本帧，用于最终结果评测。复赛将提供不同于初赛的测试集，训练集保持不变，且测试集仍包含 1382 个样本帧。示例数据可从 <https://pan.baidu.com/s/1X-1rxmZ5SQL5OnYeIPZEEA?pwd=q6px> 获取，正式数据将在报名后开放下载。

### （四）数据格式

总共采集了 10k+ 雷达点云帧和相应数量图像帧，从数据集中采集的数据中选择多个连续帧场景作为数据集样本集，并从中选取 7933 个关键帧进行手工标注。我们标注了所有对象的类、3D 边界框、旋转和 id，并将它们转换成 KITTI 标签的格式，如图 4.4 所示。带注释的目标是小型汽车 Car、骑行者 Cyclist、卡车 Truck。由于行人点云数量少且难以识别，没有对其进行标注。

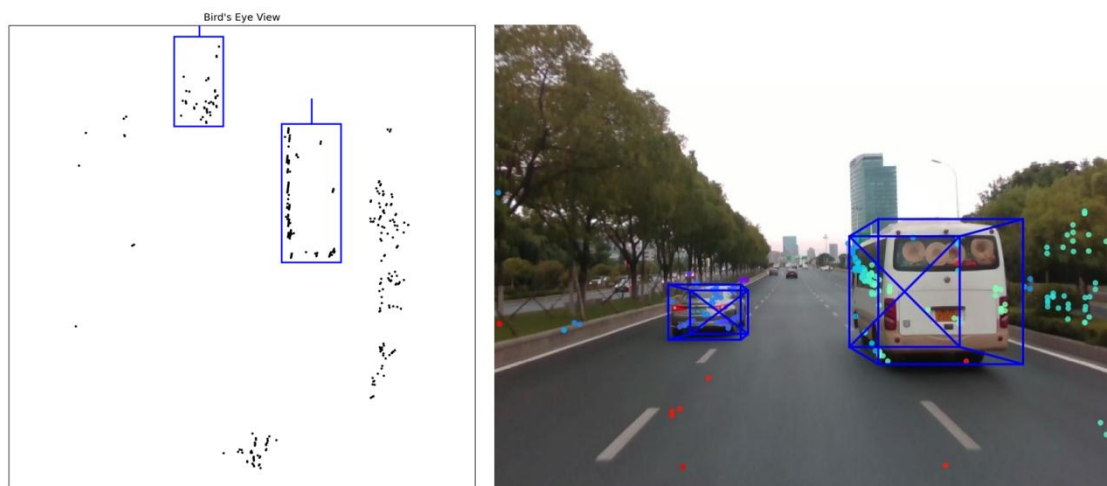


图 4.4 数据标注示意

如下图 4.5 数据集组织格式所示，DRadDataset 数据集其中 ImageSets 文件夹中包含数据集中样本集的划分。training 文件夹包含训练集和验证集数据，calib 文件夹中包含上文（二）参数标定中提到的参数标定文件。image\_2 中包含采集图像文件，label\_2 中包含手工标注的标签文件，velodyne 中包含分辨率 4D 毫米波雷达点云文件，文件格式为二进制 .bin 文件。上述图像、点云、标签文件通过统一的文件名序号关联起来。

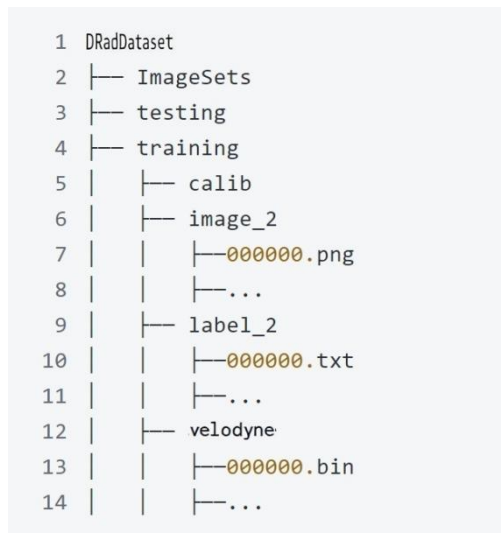


图 4.5 数据集组织格式

**calib** 文件夹说明：参考了 kitti 数据集的标定， $p_0$ - $p_3$  是相机的内参矩阵，本数据集中只用了一个相机，所以只有  $p_0$  有用， $p_1$ - $p_3$  可忽略， $p_0$  即为（二）参数标定中的内参矩阵  $K$ ，通过相机出厂参数设置的。 $r_0\_rect$  是相机的修正矩阵，近似于单位矩阵，故可忽略。 $tr\_velo\_to\_cam$  是从点云转到图像的外参矩阵，这个是根据雷达和相机摆放位置调试的。 $tr\_imu\_to\_velo$  参数为单位矩阵也可忽略。

**label** 文件夹说明：标签数据文件是 .txt 格式，每一行表示图像中一个物体的标注。每一个物体的标注数据为标准的 Kitti 数据集格式的 3D 目标检测标签格式，具体为  $[type, truncated, occluded, alpha, bbox\_left, bbox\_top, bbox\_right, bbox\_bottom, height, width, length, x, y, z, rotation\_y]$

1. **type**: 物体的类别。常见类别包括 Car、Pedestrian、Cyclist 等。
2. **truncated**: 物体的截断程度，值范围为  $[0, 1]$ ，表示物体是否部分被截断。0 表示物体完全可见。1 表示物体被截断得很严重。
3. **occluded**: 物体的遮挡程度，值范围为  $[0, 3]$ ：0 表示物体没有被遮挡。1 表示物体部分被遮挡。2 表示物体大部分被遮挡。3 表示物体完全被遮挡。
4. **alpha**: 物体的观察角度，表示相机坐标系与物体的夹角。
5. **bbox\_left, bbox\_top, bbox\_right, bbox\_bottom**: 物体在图像中的边界框坐标，分别表示边界框左、上、右、下的像素坐标。
6. **height, width, length**: 物体的 3D 尺寸（高度、宽度、长度）。
7. **x, y, z**: 物体在 3D 空间中的位置坐标，单位为米。
8. **rotation\_y**: 物体的朝向，表示物体在 3D 空间中的旋转角度，通常是绕 y 轴的旋转角度，单位为弧度。

## 五、算法设计要求

### （一）模型类型

鼓励参赛者采用深度学习算法，如基于体素或柱体的 3D 目标检测网络（如 PointPillars、SECOND）；基于点的 3D 目标检测网络（如 PointRCNN）；还有基于点与体素的方法（如 PVRCNN）。可以与视觉目标检测模型（如 YOLO、Faster R-CNN）的融合架构，也可结合 Transformer 等跨模态特征融合方法。例如：使用稀疏卷积网络（Sparse CNN）提取 4D 毫米波雷达点云的几何特征。采用 CNN 或 Vision Transformer 提取单目图像的语义特征。设计多模态融合模块（如 BEV（鸟瞰图）空间特征对齐、注意力机制融合等）。

## （二）创新性

鼓励提出创新的融合策略或改进现有算法，以提高检测精度和鲁棒性。例如，设计新的特征特征融合模块，使其能够充分提取提取 4D 毫米波雷达稀疏特征中的多尺度上下文信息，同时可以对特征进行增强，或为高分辨率 4D 毫米波雷达密集点云特征与图像特征的融合提供一种有效的解决方案以提高 3D 目标检测精度。

## （三）可扩展性

算法应具备良好的可扩展性，能够在不同配置的计算设备上运行，且在处理大规模数据时性能稳定。例如，算法应能够在普通工作站和云端服务器上高效运行，并且在增加数据量时模型性能不会出现明显下降。

# 六、性能指标要求

## （一）主要指标

1. **IoU（Intersection over Union）**：用于衡量模型检测结果与实际目标位置的重叠程度。其计算方式为目标框的交集面积与并集面积的比值。IoU 值的范围在 0 到 1 之间，越接近 1 表示模型检测结果与实际目标位置的匹配度越高。IoU 在目标检测算法的评估和优化中具有广泛应用，特别是在需要高精度目标定位的任务中，如自动驾驶、物体识别等。

2. **AP（Average Precision）**：是目标检测任务中用于评估模型性能的一项重要指标。它结合了精确率（Precision）和召回率（Recall），通过精度-召回曲线（P-R 曲线）下的面积来度量模型对不同类别的目标检测性能。AP 值越高，表示模型在相应类别上的性能越好。在多类别目标检测任务中，可以计算每个类别的 AP 值，然后取平均值得到 mAP。

## （二）次要指标

1. **模型大小**：训练得到的模型文件大小，是衡量模型复杂度和存储需求的重要指标。较小的模型大小表明模型复杂度较低，存储成本和部署难度也相对较低，更便于在不同设备和环境中应用。

# 七、功能要求

## （一）准确性

算法在检测车辆目标位置和标记 3D 边界框以及分类时，须具备高准确性，确保对包含稀少点云的骑车者目标也能精准识别，减少漏检和误检情况。对于目标车辆分类，预测结果应与真实情况高度吻合。在测试集上，3D 目标检测的交并比 IoU 值需达到[0.5]以上。

### （二）可靠性

面对不同质量、不同采集场景获取的 4D 雷达点云帧和，算法应能稳定运行，输出可靠结果。即使点云存在噪声、变化等干扰因素，算法也不应出现大幅性能波动，保持对车辆 3D 目标检测与分类的准确性和稳定性。

### （三）可解释性

算法应具备一定的可解释性，能够为自动驾驶领域相关人员提供关于车辆 3D 目标检测与分类结果的解释依据。例如，通过可视化技术展示模型在识别目标时关注的关键区域，或者提供特征重要性分析，说明模型依据哪些特征进行 3D 目标检测和分类，帮助用户理解算法决策过程，增强对算法结果的信任度。

### （四）实时性

在自动驾驶场景中，决策处理时间至关重要。算法需满足一定的实时性要求，对单例样本帧进行 3D 目标检测与分类的时间应控制在[150 毫秒]以内，以便自动驾驶系统能够快速感知理解道路情况和障碍物等环境信息从而为下游的规划决策模块提供关键的感知数据进行快速决策，提高安全性。

### （五）鲁棒性

算法要对数据的异常值、缺失值等情况具备较强的鲁棒性。在部分点云数据中存在少量错误标注、数据缺失时和杂波干扰，仍能保证检测和分类结果的可靠性，不会因为数据的小瑕疵而导致性能大幅下降。

### （六）多模态融合能力

若参赛者采用多模态数据融合方法（如结合雷达点云、视觉图像等），算法应能有效整合不同类型的数据，且在融合后能显著提升 3D 目标检测的准确性和可靠性，展现出对多源信息的高效利用能力。

## 八、开发环境

### （一）编程语言

Python，建议使用 Python3.6 及以上版本，因其具有丰富的科学计算库和深度学习框架支持。

### （二）深度学习框架

推荐使用 TensorFlow2.x 或 PyTorch1.x，这两个框架在深度学习领域广泛应用，具有高效的计算性能和丰富的 API，便于模型的搭建、训练和部署。



### （三）计算资源

参赛者可使用本地工作站或云端计算平台进行开发和训练。本地工作站需配备 NVIDIA GPU（如 GTX10 系列及以上，或 RTX 系列）以加速深度学习计算；云端平台可选择阿里云天池、腾讯云 TI 平台、百度 AIStudio 等，这些平台提供了多种配置的计算资源，方便参赛者根据需求灵活选择。

### （四）依赖库

需安装 NumPy、Pandas、Matplotlib 等用于数据处理和可视化的库；安装 Open3D、PointNet 等用于点云数据处理的库；以及安装与所选深度学习框架对应的相关库，如 TensorFlow 配套的 Keras、PyTorch 配套的 TorchVision 等。

## 九、成绩评价

### （一）输入数据格式要求

参赛者的算法应能正确读取主办方提供的类似当前被广泛用作自动驾驶相关算法训练与评测的 KITTI 数据集的数据组织形式 DRadDataset 数据集。对于其中的 png 格式的图像文件，需能够解析其中包含的图像信息。对于标签文件，需能够准确提取由手工标注的所有对象的类、3D 边界框、旋转和 id 信息。对于 calib 文件夹中的标定参数文本文件，需能够准确提取 4D 毫米波雷达与图像之间的关系矩阵信息。对于 velodyne 文件夹中包含的高分辨率 4D 毫米波雷达点云文件，格式为二进制 .bin 文件，需能够准确解析并提取其中的信息。

### （二）输出数据格式要求

输出格式要求与训练数据集标签文件夹 label\_2 格式一致，其中包含对每个测试样本的预测文件 (.txt)。每个预测文件需与对应的输入数据文件名一致（如 000001.txt 对应 000001.png/000001.bin）。

每个预测文件每行包含一个检测到的物体，若无检测目标，则文件为空。每行包含 16 个字段，字段间以空格间隔，顺序固定如下：

字段序号	字段名称	数据类型	单位/范围
1	类别	字符串	Car/Cyclist/Truck
2	截断程度	浮点数	[0, 1]
3	遮挡等级	整数	0/1/2/3
4	观测角度	浮点数	$[-\pi, \pi]$
5-8	2D 边界框	浮点数	像素坐标
9-11	3D 尺寸	浮点数	米
12-14	3D 中心坐标	浮点数	米
15	旋转角	浮点数	$[-\pi, \pi]$
16	检测置信度	浮点数	[0, 1]

### 示例文件内容：

以检测到一辆车和一个骑行者为例

```
Car 0.00 0 1.57 712.40 143.00 810.73 307.92 1.65 1.67 3.64 -5.14 2.78 9.12 1.56 0.95
Cyclist 0.30 1 -0.50 512.30 180.40 550.21 220.70 1.75 0.60 0.80 2.40 1.89 15.20 -0.30 0.80
```

### （三）成绩计算公式

成绩将根据算法输出结果与真实标注数据对比，依据性能评估指标（如 IoU、mAP 值等）加权求和计算进行打分…

### （四）有效成绩

最终成绩评分高于 40 为有效成绩，阈值 40 确保目标检测算法具有实际应用价值。赛题设奖基数为有效成绩团队数量。

## 十、解题思路

### （一）数据预处理

对点云数据进行滤波处理，去除背景噪声点，保留目标前景点（如机动车，行人，骑行者等），减少无关点的干扰，增强目标区域的点云密度。然后将稀疏的点云划分为规则的 3D 体素网格，形成结构化的表示，每个体素是一个固定大小的立方体，降低计算复杂度，同时保留空间几何信息。对图像采用数据增强技术，如旋转、缩放、翻转等操作，扩充 训练数据集，增强模型的泛化能力。还可以对图像像素值进行归一化处理。

### （二）特征提取

利用稀疏卷积处理体素化后的点云，提取 3D 稀疏体素特征，设计不同的稀疏卷积层、池化层组合，使在处理 3D 稀疏数据时能够显著降低计算复杂度，提高计算效率。对于视觉图像处理，通过卷积神经网络强大的特征提取能力，捕获高级语义信息。将点云和视觉图像都映射到 BEV 空间进行融合，生成统一的 BEV 融合特征，简化了高度信息，降低了计算复杂性，同时保留了尺度一致性和 3D 几何信息。

### （三）模型训练

选择合适的深度学习框架（如 TensorFlow、PyTorch）搭建 模型，设置合理的训练参数，如学习率、迭代次数、批量大小等。在训练过程中，采用交叉验证方法，利用验证集数据对模型进行评估和调优，防止模型过拟合。

### （四）模型融合与优化

可尝试将多个不同结构或训练阶段的模型进行融合，如采用投票法或加权平均法，综合多个模型的预测结果，提高最终预测的准确性。同时，根据性能评估指标，对模型进行针对性优化，如调整模型结构、增加训练数据量等。

## 十一、参考资源

### （一）书籍

1. 《深度学习》（DeepLearning），由 IanGoodfellow、YoshuaBengio 和 Aaron Courville 撰写，系统介绍了深度学习的基础概念、模型架构和训练方法，对理解和应用神经网络有很大帮助。

2. 《Python 深度学习》（DeepLearningwithPython），作者 FrançoisChollet，通过大量代码示例，详细讲解了如何使用 Python 和 Keras 框架进行深度学习模型的开发，适合初学者快速上手。

### （二）在线课程

1. Coursera 上的“DeepLearningSpecialization”课程，由吴恩达教授授课，涵盖了深度学习的多个关键领域，包括神经网络基础、卷积神经网络、循环神经网络等，课程内容丰富且实践性强。

2. edX 上的“IntroductiontoArtificialIntelligence”课程，提供了人工智能和机器学习的入门知识，包括算法原理、模型训练和应用案例等，有助于参赛者构建全面的知识体系。

### （三）学术论文

在学术数据库中搜索关于 3D 目标检测的最新研究论文，如“PointRCNN: 3D object proposal generation and detection from point cloud”等，了解当前该领域的前沿技术和研究方法。

关注知名人工智能会议（如 AAAI 等）上发表的相关论文，跟踪最新的研究动态和创新成果。

## 十二、提交要求

### （一）算法代码

提交完整的算法代码，包括数据预处理、模型训练、预测推理等各个环节的代码。代码需使用 Python 语言编写，并遵循 PEP8 编程规范，具备清晰的注释和文档说明，以便评审人员理解和运行。

### （二）技术报告

提交详细的技术报告，内容包括算法设计思路、模型架构图、实验设置（如训练参数、数据增强方法等）、性能分析（对主要指标和次要指标的详细分析）以及算法的创新点和不足之处。技术报告格式采用 PDF，字数不少于[3000]字。

### （三）模型文件

提交训练好的模型文件，并提供模型的加载和使用说明，包括所需的运行环境、依赖库等信息。模型文件应能够在指定的测试环境中正常运行并输出预测结果。

## 十三、比赛流程及奖项设置

### （一）报名阶段

参赛者在比赛官方网站上完成报名注册，提交个人或团队信息，获取初赛数据下载链接。

### （二）初赛阶段

参赛者利用赛事方提供的训练数据集进行算法模型设计，利用提供的初赛测试集进行相应方法的验证与调试。初赛阶段参赛者每天提交结果的次数不限，但是初赛排行榜每隔 1 小时刷新一次。

### （三）复赛阶段

初赛结束后进入复赛阶段，开放复赛数据下载链接。仅有初赛阶段提交有效结果的参赛团队可以进入复赛。复赛期间，参赛者利用赛事方提供的复赛阶段数据进行算法模型调试，提交对复赛测试数据的推理结果。复赛阶段持续 3 天，每个参赛队伍每天仅能提交 2 次。复赛排行榜每隔 1 小时刷新一次。

### （四）复赛成绩公布

在比赛官方网站上公布复赛成绩。以进入复赛参赛团队数量作为计奖基数，按照不超过大赛省赛设奖比例，评选出复赛一、二、三等奖（颁发省赛获奖证书）。评选复赛奖过程中，参赛者提交的算法性能低于赛事方提供的基线参考分数的判定为无效成绩，不予授奖。复赛一、二等奖晋级参加国赛总决赛。

### （五）决赛阶段

1. 决赛线上评选。晋级决赛的参赛团队，依据复赛排行榜结果，以进入决赛参赛团队数量作为计奖基数，按照不超过大赛国赛设奖比例，评选出国赛一等奖候选名单及国赛二、三等奖获奖名单（颁发国赛二、三等奖证书）。

2. 决赛作品提交。国赛一等奖候选团队在规定时间内提交技术文档、算法代码和模型文件、演示视频、补充材料等。提交截止后，不再接受任何形式的修改和补充。

3. 决赛审核阶段。由专业评审团队对国赛一等奖候选参赛团队的参赛作品进行结果复现与审核。评审过程中如有疑问，可要求参赛者进行解释说明。

4. 决赛线下答辩。国赛一等奖候选团队在规定时间内提交完善后的技术文档、算法代码和模型文件、演示视频、补充材料，参加国赛线下总决赛复核答辩，最终依据算法性能得分和线下答辩得分确定国赛一等奖获奖名单及其排名（未参加线下复核答辩视同放弃奖项）。国赛一等奖颁发荣誉证书。

## 十四、其他说明

### （一）公平性

严禁任何形式的作弊行为，包括但不限于数据泄露、模型预训练数据与测试数据重叠、抄袭他人代码等。一经发现，立即取消参赛资格，并追究相关责任。

## （二）知识产权

参赛者提交的作品必须为原创，未在其他比赛中获奖或公开发表。比赛主办方有权对参赛作品进行展示、宣传等相关活动，但知识产权仍归参赛者所有。

## 十五、联系方式

赛项交流 QQ 群：730575503

邮箱：ymhhmy01@foxmail.com