「深度学习」在毫米波雷达感知中的应用

Original 苏煜 焉知新能源汽车 2021-06-09 11:40

收录于合集

#智能驾驶 43 #自动驾驶 35 #毫米波雷达 2 #传感器 4

作者言：

由于工作的关系，一直关注自动驾驶技术中的传感器感知算法，平时会读相关的论文，跟踪学术界和工业界最新的进展。

自动驾驶是近些年来非常火热的方向，感知技术也是日新月异的发展，因此有必要系统性的梳理技术的脉络，一方面方便自己随时查阅，另一方面也期望和同道中人多多交流。

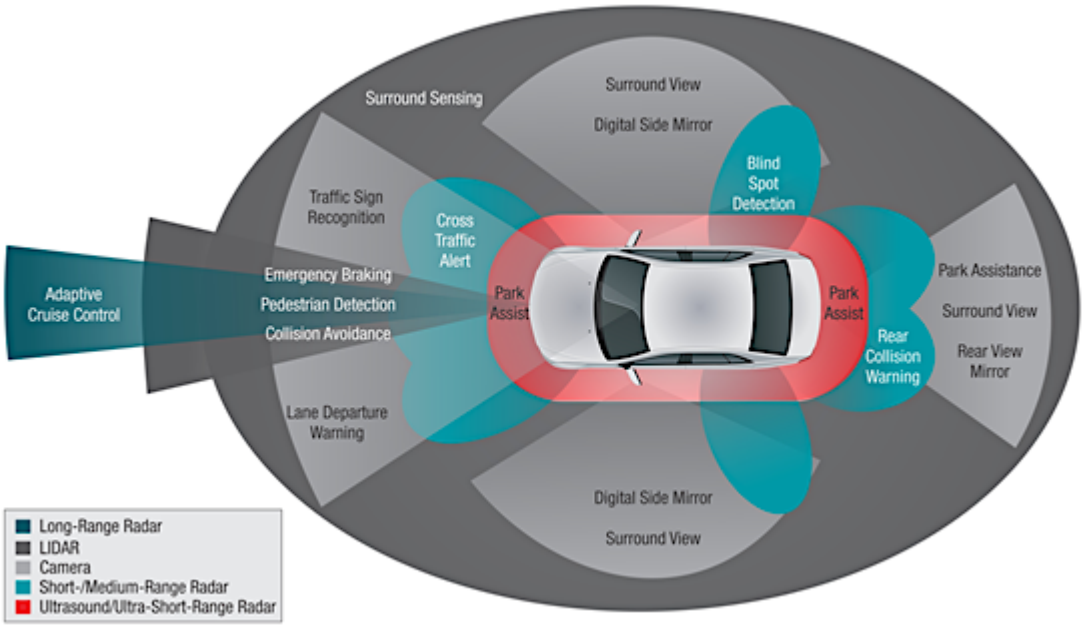
自动驾驶是一项复杂的技术，其核心包括传感器感知、行为决策、高清地图，海量数据、高性能计算平台等等。

目前，基于深度学习的方法在传感器感知方面取得了突破性的进展，并因此推动了自动驾驶技术的飞速发展，自动驾驶技术所采用的传感器主要包括摄像机、激光雷达和毫米波雷达。

摄像机用于采集可见光图像，对于物体的形状和类别的感知精度较高。由于深度学习技术的成功起源于计算机视觉任务，**很多成功的算法也是基于对图像数据的处理，因此基于摄像机的感知技术目前已经相对成熟。**图像数据的缺点在于缺少了场景和物体的距离信息，且受天气和环境的影响较大。

激光雷达在一定程度上弥补了摄像机的缺点，可以精确的感知物体的距离，但是限制在于成本过高，难以大批量生产。

毫米波雷达具有天线波束窄、分辨率高、频带宽、抗干扰能力强等点，可以比较精确的测量物体的速度和距离，受天气和环境影响较小，而且成本较低，易于大规模生产。因此，也成为了目前自动驾驶技术研究的一个热点方向。



多种传感器及其在自动驾驶中的应用

目前，基于摄像机和激光雷达的深度学习算法研究的相对比较全面，相关的综述文章也比较多。因此，本专栏会首先介绍研究相对较少的毫米波雷达感知算法。传统的雷达感知技术包含大量基于规则的算法。这些规则是研究和工程人员利用先验知识和实际数据总结出来的。

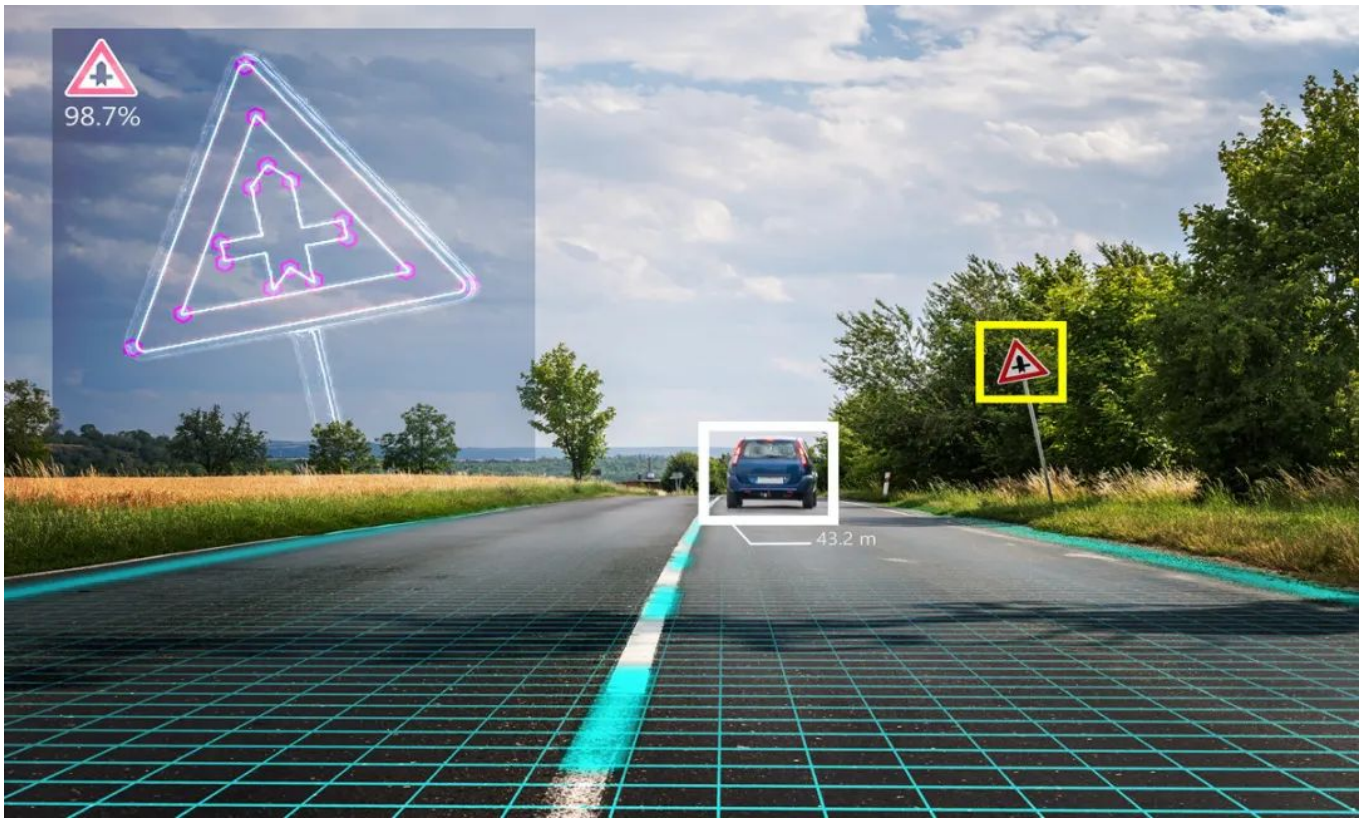
人类的先验知识是有限的，可处理的数据也是有限的，因此这些规则并不完备，无法涵盖实际场景中的所有情况，而深度学习技术可以自动的从海量数据中获取知识和规则。**随着数据不断累积，知识和规则的学习也就越来越完备，并超越基于人工规则的算法。**因此，近些年来，学术界和工业界都都在积极的探索如何将深度学习技术用于提高雷达感知算法的性能。

01

毫米波雷达信号处理

雷达信号与图像信号有很大的差别（与激光雷达部分相似），所以在介绍具体的算法之前，有必要先简单了解一下雷达信号长的什么样子。

雷达的种类非常多，**比如连续波雷达、脉冲雷达、相控阵雷达、合成孔径雷达（SAR）等。**不同雷达应用的场景也不尽相同，本文只关注自动驾驶应用中常用的毫米波雷达。

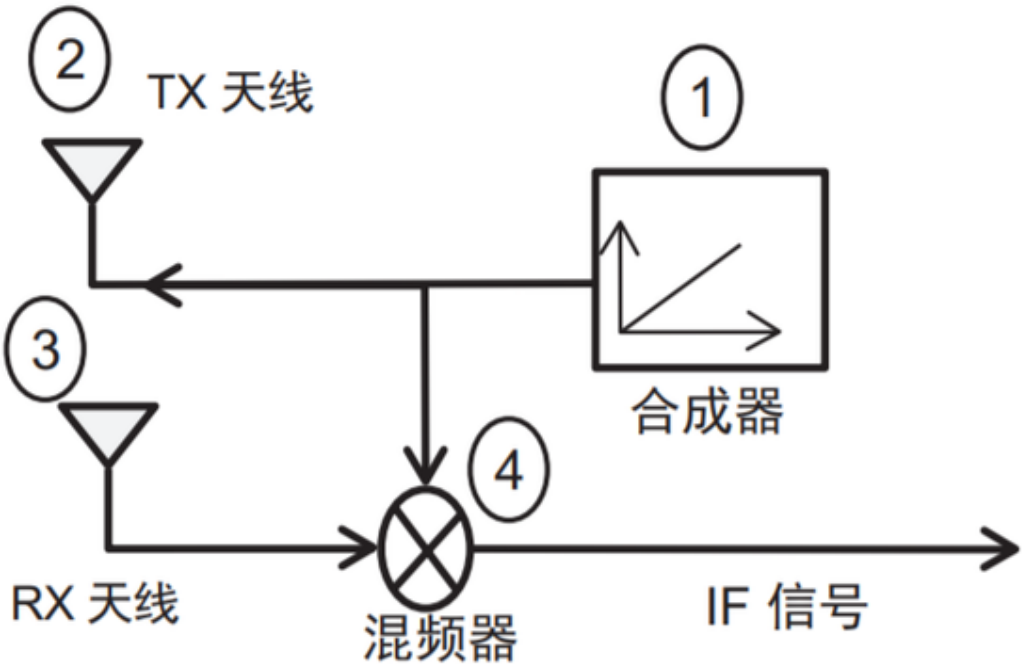


毫米波雷达发射波长为毫米量级的信号。这种短波长一方面可以使天线的尺寸做的很小，另一方面可以提高检测的准确度，比如工作频率为 76 - 81 GHz（对应波长约为 4 mm）的毫米波雷达可以检测零点几毫米的移动。

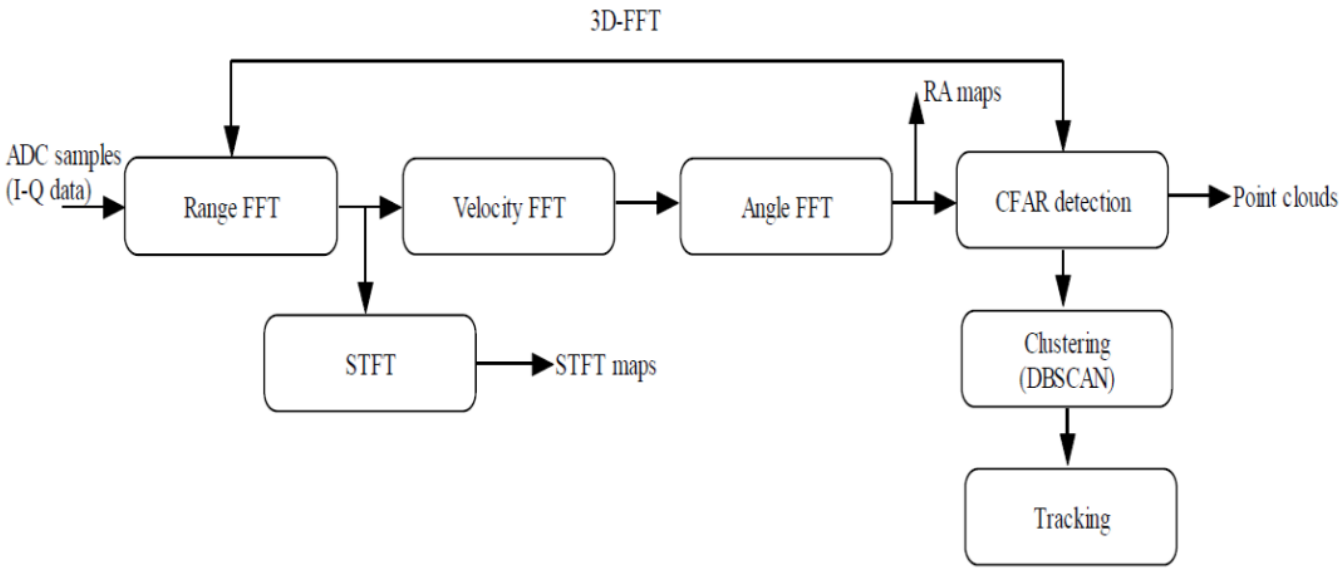
目前，**常用的是一种称作调频连续波 (FMCW) 的特殊毫米波技术。FMCW 雷达连续发射调频信号，以测量目标的距离、角度和速度。**

简单来说，其工作原理如下。首先，合成器生成一个线性调频脉冲， 该线性调频脉冲由发射天线（TX 天线）发射。目标物体对该线性调频脉冲的反射生成一个由接收天线（RX 天线）捕捉的反射线性调频脉冲。混频器将 RX 和 TX 信号合并到一起，生成一个中频 (IF) 信号。

一般来说，一个雷达包含多个发射和接收天线，因此也就会得到多个 IF 信号。目标物体的信息，比如距离、速度、角度都包含在这些 IF 信号中。通过对 IF 信号进行多次离散傅里叶（DFT）变换，即可将这些信息分离出来。



FMCW 雷达结构图



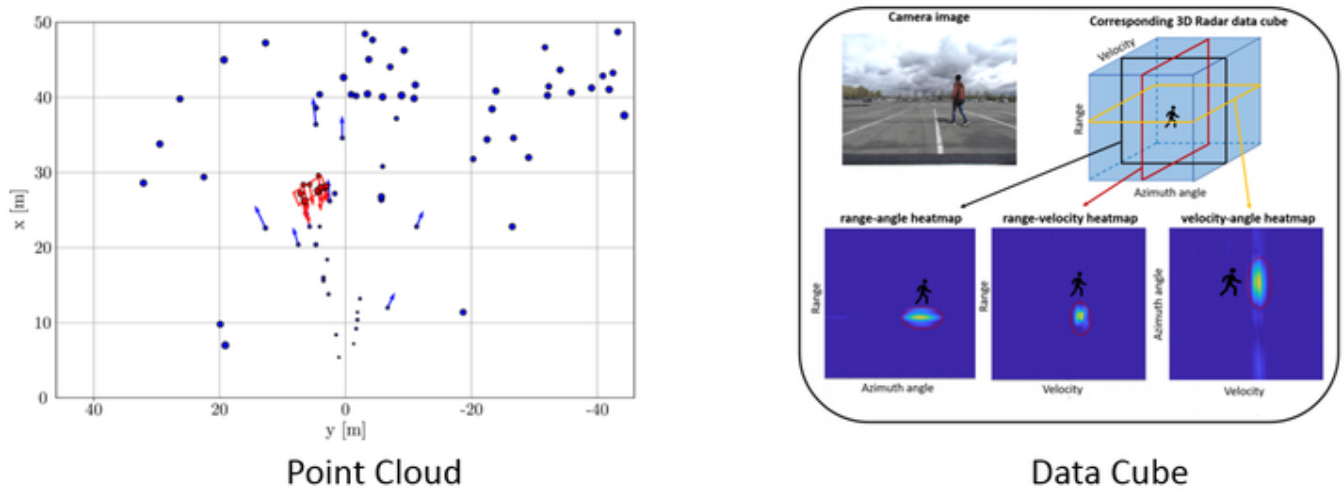
传统的雷达信号处理流程

ADC 信号经过三次 DFT 处理之后得到一个离散但是稠密的三维信号，维度分别对应 Range（距离）、Angle（方向）和 Doppler（速度）。传统的雷达信号处理会采用一个称作 CFAR（Constant False-Alarm Rate）的算法来对信号进行过滤，只保留比较强的响应。

CFAR 算法的关键就是动态地确定采样的阈值，**CFAR 采样后得到的稀疏的数据又被称为点云（Point Cloud）。每个点的数据包括 Range、Angle、Doppler 以及 RCS（可以粗略的理解为目标的反射面积）。**

CFAR 算法设计简单计算量小，可以过滤掉大量噪声，但也不可避免的会丢失有用的信息。另外一个选择是不采用过滤算法，直接保留所有的稠密数据（三维的张量），我们可以将其理解为一个数据块（Data

Cube）。当然我们也可以采取折中的办法，也就是降低阈值，保留更多的数据点，**也将其视为一个数据块，只不过这个数据块是相对稀疏的。**



两种不同的雷达数据表示方式

把雷达数据的表示形式分为 Point Cloud 和 Data Cube 两种，是因为处理这两种数据会用到截然不同的深度学习算法。下面我们一个一个看。

02

Point Cloud 数据的处理算法

雷达的点云是一种非常稀疏的信号，比如一辆汽车上一般来说只有几个到几十个数据点，一个行人上可能只有几个数据点。相对于稠密的数据块来说，点云的的数据量非常少，因此算法很相对轻量，对硬件的要求较低，通常会应用在较低成本的感知系统中，作为其他传感器的补充。

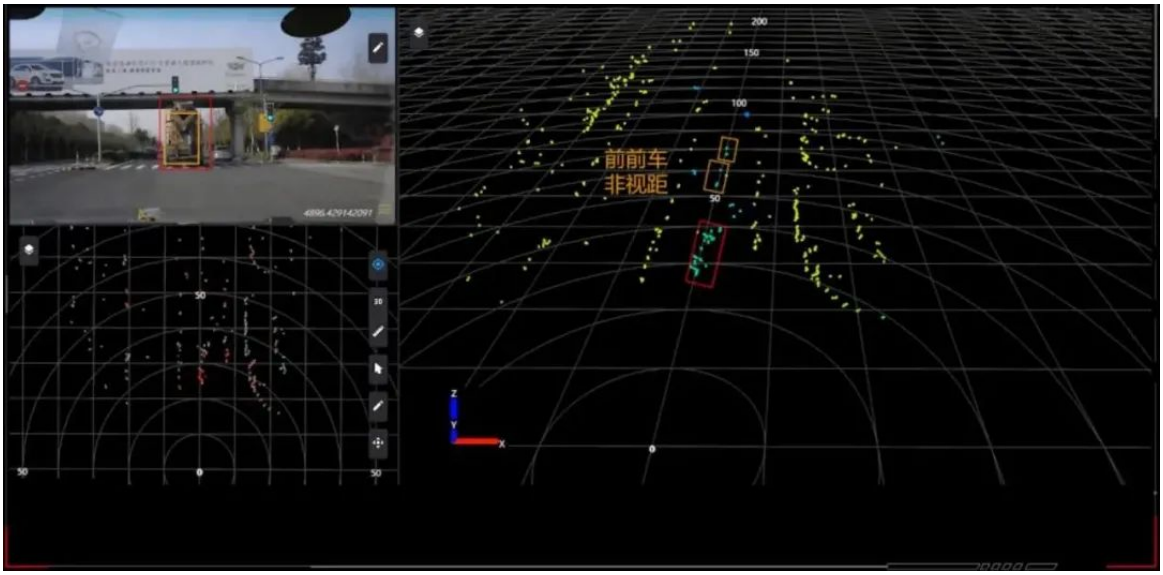
在算法方面，大部分的工作都是借鉴了激光雷达点云的处理方式。这些方法大致可以分为两种：**一，直接处理点云数据，比如利用 PointNet/PointNet++；二，将点云转换为网格数据，从而采用卷积神经网络（CNN）来处理。**

直接处理点云

早期的工作一般采用聚类的方式得到目标物体的Proposal，然后提取手工设计的特征，最后输入给分类器进行处理。下面是这个方向的一个代表性工作。

此前有戴姆勒的工程师采用 DBSCAN 来对点云进行聚类，**对每个 Cluster 提取 34 维特征。分类器对比了 Random Forest 和 LSTM，其中 LSTM 的输入为 8 个连续的 34 维特征向量。分类器输出为六个类别，包括车辆、行人、垃圾桶等。**

实验结果表明，LSTM 并没有很大的优势。原因之一是序列长度不够，这个受限于其采集的数据。另外手工设计的特征也限制了 LSTM 的学习能力。



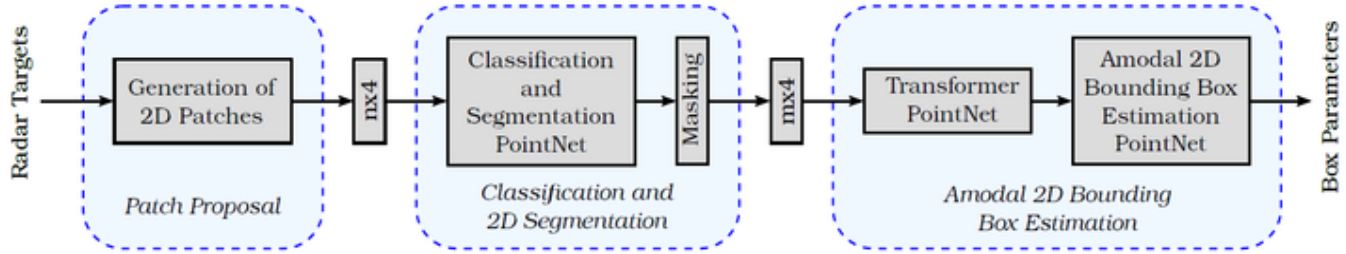
众所周知，深度学习的一大优势在于可以自动的特征学习，**因此研究者逐渐放弃了人工设计特征的方式，转而将此任务交给神经网络，并逐渐过渡到端对端的学习方式。**下面是这个方向两个典型的工作。

来自德国 Ulm 大学的工程师做了一个实验，借鉴了 Frustum PointNets 方法，其基本思路可以理解为：**物体检测中常见的 two-stage 方法。**

首先生成 object proposal，这里直接将每个点看做一个 Proposal，Region 的大小根据物体的先验知识来确定。每个 Proposal 包含 n 个点，每个点包括 x、y、Speed 以及 RCS 四个特征。

接着利用 PointNet 或者 PointNet++ 对 Proposal 进行分类，也对 Proposal 中的每个点分类，称为**点分割任务**，并过滤掉背景点。最后只对事物体的 Proposal 进行 boundingbox 预测。

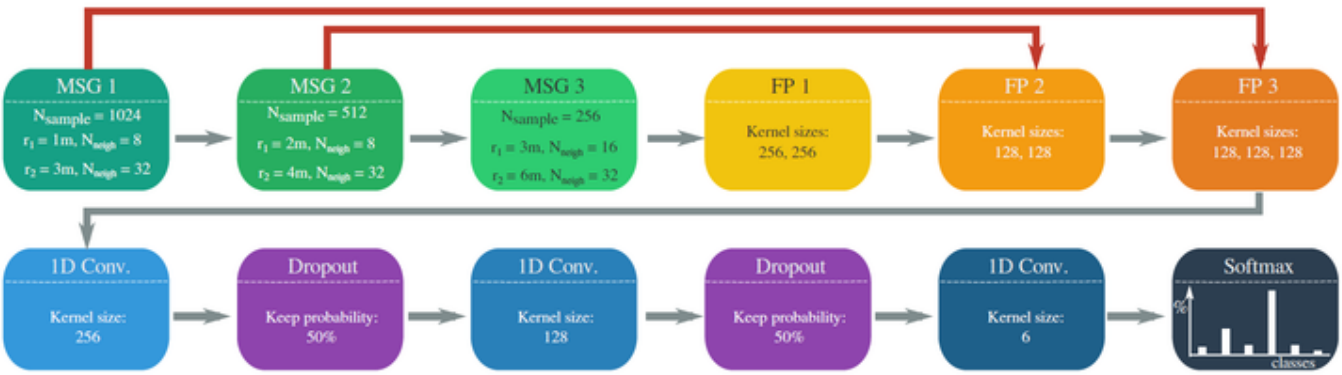
但这个实验是在封闭环境下进行的，只有一辆目标车辆，Ego 车辆也只一个 Radar sensor，因此并不具备太多参考价值。



同样来自戴姆勒的工程师，采用基于 PointNet++ 的方法对点云进行语义分割，也就是**每一个点分配一个类别标签（比如车辆、行人、背景等）**。

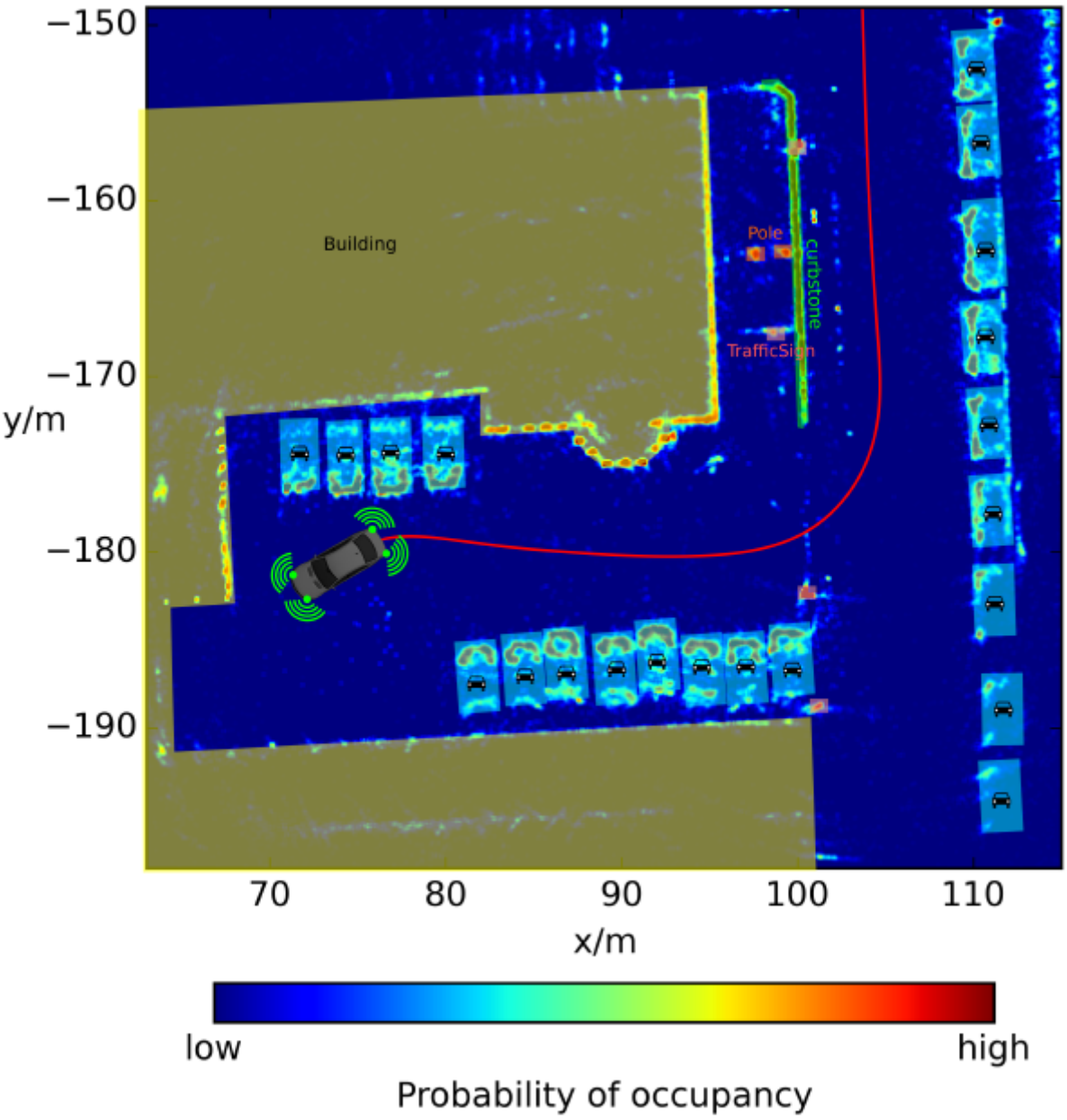
值得一提的是，他在时序上对多帧的点云进行融合，以提高点云的密度，这是其成功的关键点之一。另外，工程师利用了多尺度的邻域信息，并用一种类似 U-Net 的方式连接起来。

数据方面，**内容采用了多个 Radar sensor，并且在真实路况下采集数据，人工标注每个点的类别。**



点云转换成网格数据

在早期阶段，研究人员一般会基于多帧的点云信号来生成 Occupancy Grid，用来检测静态障碍物。Occupancy Grid 与 2D 图像的结构相同，可以直接用 CNN 来进行物体检测和分类。比如下面这个工作就是在 Occupancy Grid 上复制出 8 m x 8 m 的候选区域，然后用 CNN 对这些区域进行分类。



基于 Occupancy Grid 的静态物体分类

Occupancy grid 只能表示静态物体，因此更为一般的方法是将点云量化为 2D 的网格结构，每个网格内可能包含 0 到 N 个点。将多个点的数据转化成一个定长的向量，就得到了一个 HxWxC 的 3D Tensor 数据，其中 H 和 W 表示网格的维度，C 表示特征向量的长度。

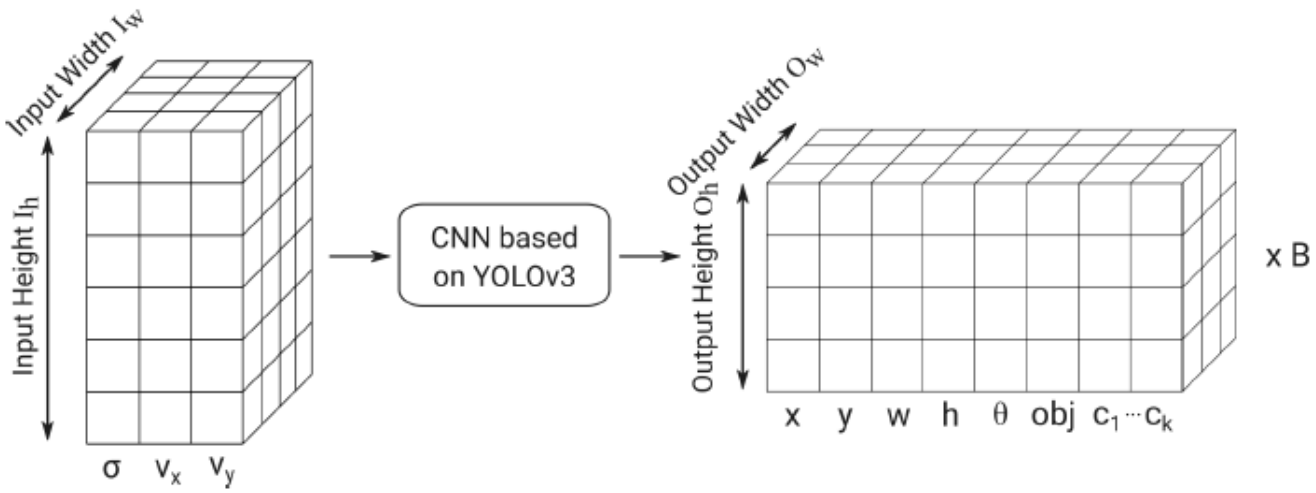
这是一个可以被 CNN 处理的标准数据类型，各种基于 CNN 的物体检测网络都可以用来执行检测任务。这里比较特别的一步就是如果将多个点转换为定长的特征向量，**可以采用简单的 Mean/Max Pooling，也可以采用类似于 VoxelNet 中的 Voxel Feature Encoding 的方法。**VexelNet **是针对于 Lidar 的 3D 点云设计的网络，也可以魔改一下用于 2D 的 Radar 点云。**

慕尼黑大学的研究者对比了基于网格和点云的方法：

基于网格的方法就是简单的将点云转换成网格，特征向量为 x 和 y 方向的速度，以及 RCS，然后利用 YOLOv3 网络得到物体的信息（类别、BBox 位置、大小和方向）。

基于点云的方法就是之前介绍的 Frustum PointNets。在 NuScences 数据库上对这两种方法进行了对比，基于网格的方法的 mAP 为 7.87%，远低于基于点云方法的 19.61%，但是网格方法在速度上（6.27 f/s）比点云方法（0.27 f/s）快很多。

值得一提的是，**NuScenes 数据库是第一个拥有同步的可见光图像，激光雷达点云和毫米波雷达点云，并且在真实路况下采集的大规模数据库。后面我会再详细介绍这个数据库，它的出现对于基于雷达点云的物体检测研究是一个很大的推动。**



基于网格数据和 YOLO 网络的物体检测

小结：直接处理点云的方法性能较好，**但是由于数据的不规则，很难利用并行计算进行加速，**因此在效率上反而不如网格方法。但是总的来说，**两种方法在真实路况数据上的效果都不太理想。**

原因大致有两点：

一，没有采取有效的时序融合，单帧的雷达检测结果包含很多噪声，传统的雷达物体跟踪算法也在很大程度上依赖于时序信息。循环神经网络（LSTM 或 GRU）可以更好的处理时序信号，所以应该是下一阶段研究的重点。

二，点云过于稀疏，很多有用的信息已经在预处理阶段丢掉了。因此才有研究者提出直接从雷达原始信号出发，采用端对端的方式来进行物体检测。

这样做的好处是充分利用深度神经网络强大的特征学习能力，自动的去除原始信号中的噪声，保留有用信息。**但是这部分网络的设计需要更多的考虑到雷达自身的特性，因此更具有挑战性。**下一部分将会介绍这方面的工作。

Data Cube 数据的处理算法

如前所述，原始的雷达信号经过 3 次 FFT 处理之后得到以 Range-Doppler-Azimuth 为坐标的 Data Cube，对这个 3D tensor 做 CFAR 处理，就得到了稀疏的点云数据。

这个过程不可避免的会丢失有用信息。随着自动驾驶芯片计算能力的不断提升，直接利用 Data Cube 作为输入数据也逐渐成为了可行的方案。近两三年来，这方面的工作慢慢多了起来，也成为毫米波雷达感知领域新的研究热点。下面就介绍几个典型的工作。

在 2019 年的来自 Qualcomm 的一篇文章《Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors 》是早期的一个比较全面的介绍了毫米波雷达的数据处理算法。

作者提出了两种处理 Range-Doppler-Azimuth Tensor 的方式：**一，是沿着 Doppler 维度将 Tensor 压缩为 2D RA tensor；二，是分别沿着三个维度压缩得到 RA、RD、AD，三个 2D Tensor，最后再合并为 RA tensor。**

这里说的压缩其实就是相加的操作，**不论采取哪种方式，最终都会得到一个以 RA 为坐标的特征图，特征图的 Channel 维度编码了 Doppler 和 Energy 信息，然后一个关键的步骤就是将这个特征图从 RA 坐标（也就是极坐标）转换到笛卡尔坐标。**

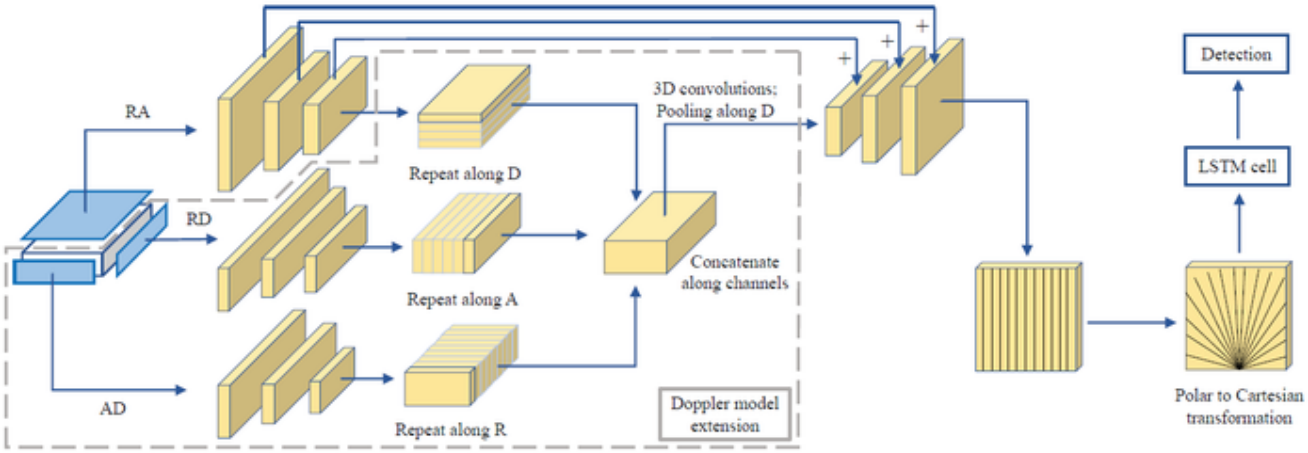
这样做的原因是在极坐标下物体的大小是随着位置变化的，这不符合卷积操作平移不变性的假设。当然坐标转换也不能完全解决这个问题，不同位置处的物体的 Signature 还是会有差别。这个问题比较复杂，留待之后再讨论。

最后，为了利用时序信息，LSTM 单元会作用在坐标转换后的特征图上。SSD Object Detector 被用来进行物体检测任务。这一步不是很关键，任何 Object detector 理论上都可以用。

实验方面，作者在高速公路的场景下采集并标注了一个中等规模的数据库（~100 K 帧，390 K 个标注）用来训练和测试。**这个数据库场景单一，物体大部分应该是移动的车辆，因此难度相对较低。**

实验结果表明：**「在没有 LSTM 的情况下，RAD 模型的性能优于 RA 模型，在有 LSTM 的情况下则基本没有差别。」**这也说明 LSTM 会自动的学习到一定的运动信息。

此外，作者还与 Lidar 算法进行对比，得出结论：**「基于雷达的检测模型受距离影响较小。」**当然这个对比没有采用同样的数据库，而且雷达数据库只包含了运动物体，因此这个结论并不是非常可信。

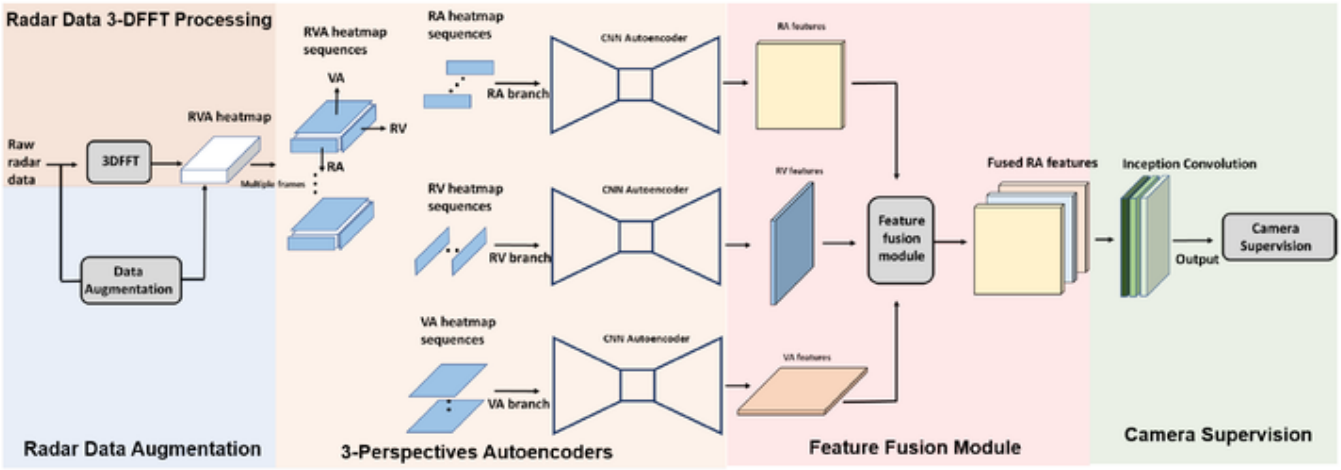


总的来说，很多方法与 Qualcomm 的很相似，都是将 RAD tensor 分解成三部分分别处理，再合并到一起。

差别主要以下几点：

- 对于有些对 RA 的处理更加复杂，RA tensor 包含了复数值，并且只在任意选择的chirp上进行（并不是完整的Doppler spectrum）；
- 利用 3D 卷积处理多帧数据；
- 用 AutoEncoder 来提取特征；
- 对于融合后的特征图，采用了 3D Inception 结构，以覆盖不同尺度的感受野。

写在最后



以上实验都是直接输入底层的雷达数据，利用神经网络进行端对端的学习来检测场景中的物体，**个人觉得这应该是未来雷达感知算法的发展趋势。**

另外还有一些方法，会先检测场景中的 ROI（类似于点云）作为候选目标，然后再用神经网络对候选目标进行分类。这类方法的缺点是无法进行端对端的学习，不能充分利用神经网络的优势，所以这里就不做介绍了。

参考

1. ^https://training.ti.com/node/1139153?context=1128486-1139153
2. ^https://ieeexplore.ieee.org/document/8126350
3. ^https://arxiv.org/abs/1904.08414
4. ^https://ieeexplore.ieee.org/document/8455344
5. ^https://ieeexplore.ieee.org/document/7918863
6. ^https://ieeexplore.ieee.org/document/9294546
7. ^https://www.nuscenes.org/
8. ^https://ieeexplore.ieee.org/document/9022248
9. ^https://arxiv.org/abs/2011.08981

蔚来发布挪威战略，2022 年进入欧洲 5 个国家，目标销售 10 万台

蔚来第 10 万辆车下线，是中国汽车实现「品牌自信」的里程碑

大众大踏步进入电动化，加速垂直整合动力电池