Deep Understanding Tesla FSD

teng.kang (康腾)

研发部/开发管理部/量产自动驾驶研发部

2021.12.18

THE FUTURE OF POSSIBLE

目录

contents

01

Freespace综述

Freespace综述

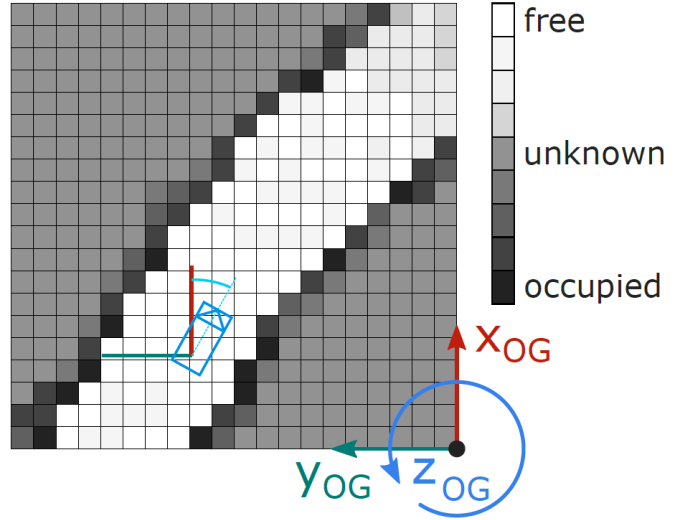
表征形式：

矢量包络表征

极坐标系下的360或者720个点，多数情况视觉使用，好处是msgsize够小，缺点是不能表达遮挡物后面的可通行情况

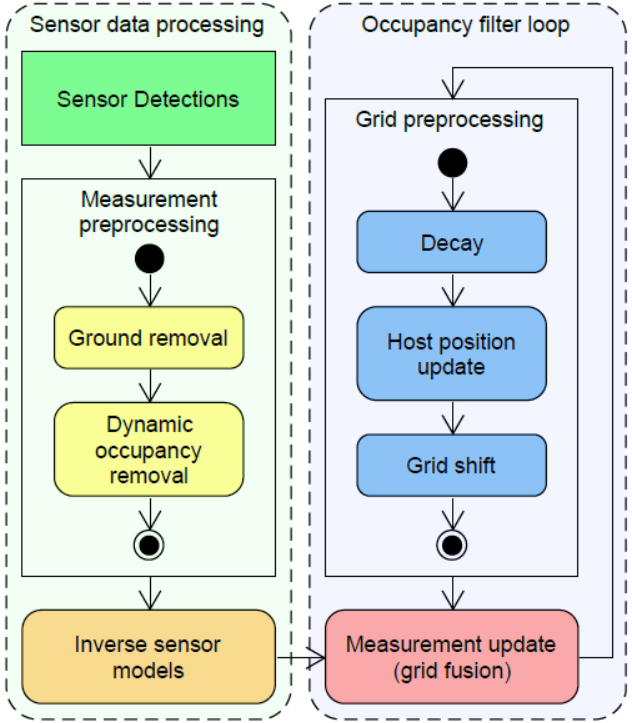
栅格表征

多激光使用，数据量大，但表征丰富，可以表达遮挡物后的可通行情况，用于复杂的规划a星搜索。

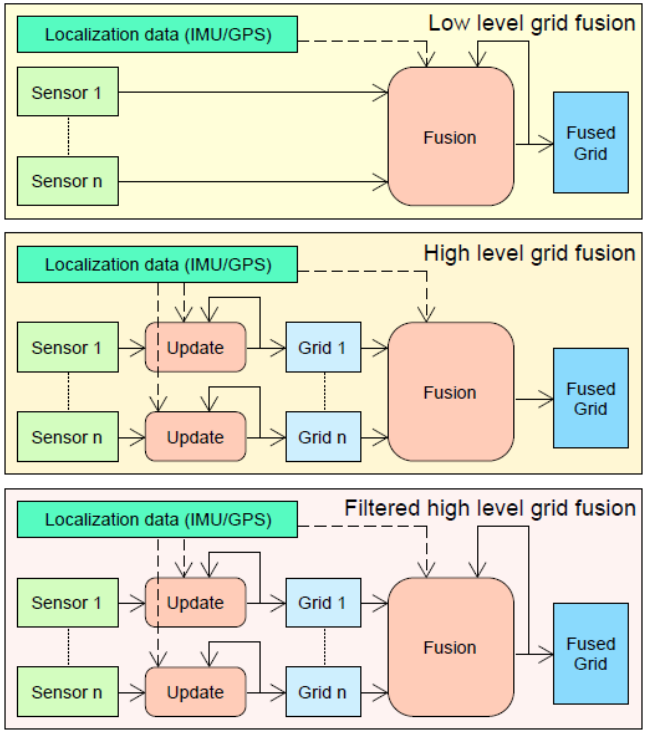


工程上，12一般都是矢量表征，14一般用栅格表征的方案

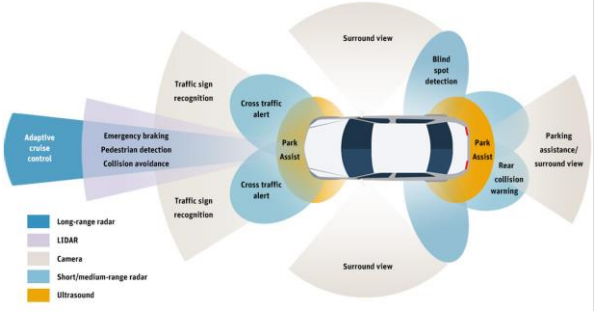
Freespace综述



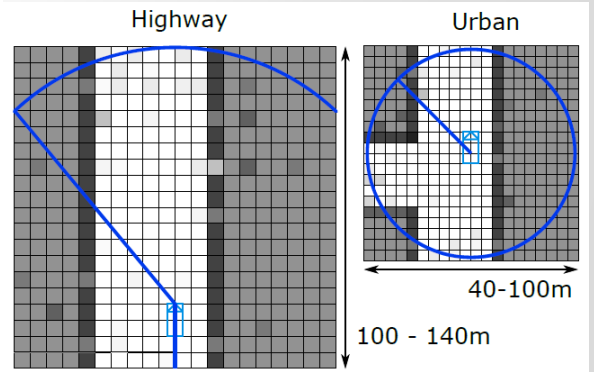
Occupancy grid algorithm flow



Low and high level grid fusion frameworks comparison



Multi Sensors



Highway and urban grid

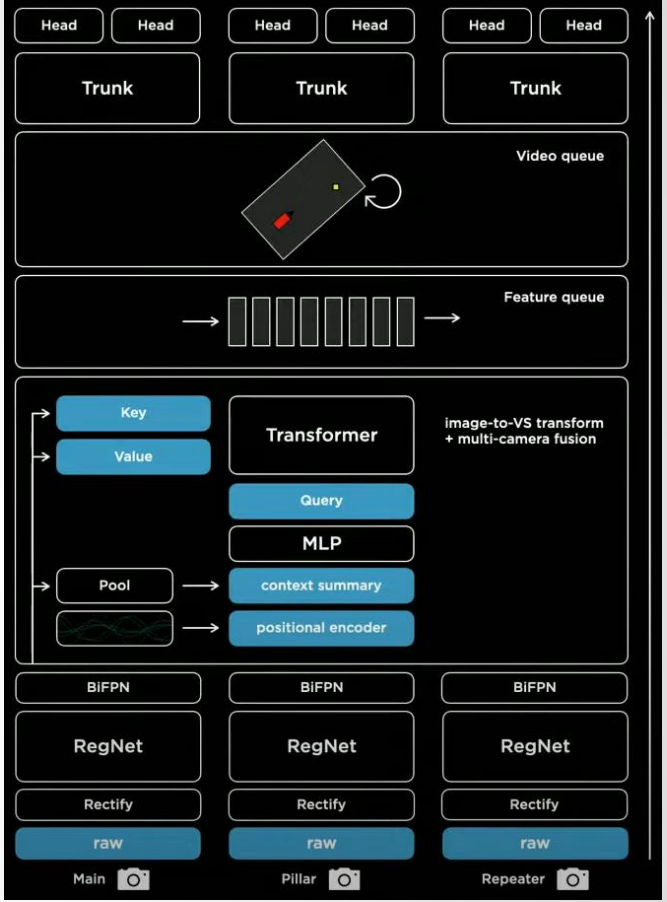
02

Tesla FSD解析

Tesla FSD解析

亮点

- Vector Space 向量空间 -- 表征形式
- Multi Camera Fusion 多相机融合 -- 空间信息
- Video Fusion 时序融合 -- 时序信息



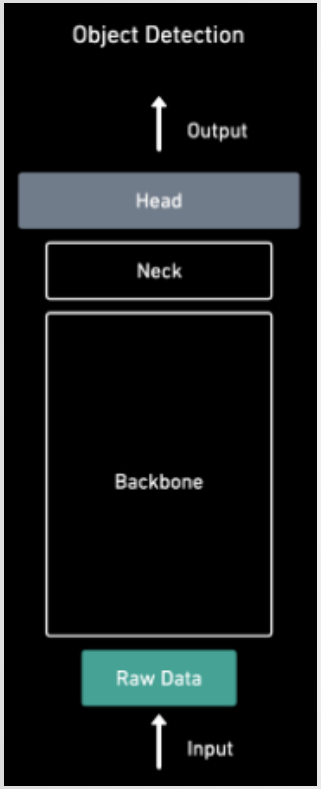
输入

- 1280x960 12 bits(HDR) @ 36Hz，RAW
- 输入使用了比较低像素的相机。
- 使用12bits HDR，应该是用来解决相机曝光问题。
- 使用RAW结果，减少马赛克算法预处理，理论上提供更多的信息量。

网络结构（HydraNet）

- Backbone： RegNet，目前主流最好的backbone
- Neck： BiFPN，多尺度的传统做法
- Heads： HydraNet Detection:

Tesla FSD解析



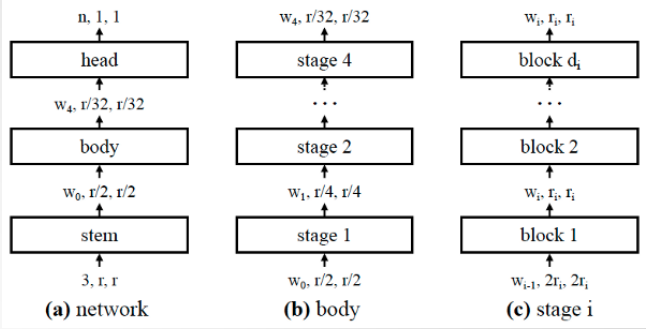
Object Detection Structure

backbone: RegNet

2020 Facebook

Designing Network Design Spaces

1. 一个非常好的设计空间
2. 权衡延迟和准确性



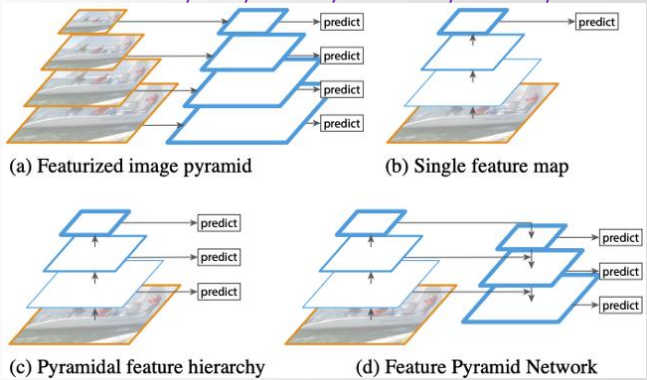
AlexNet/VGG/ResNet/DenseNet/AutoML/NAS

neck: BiFPN

2019 Google

EfficientDet: Scalable and Efficient Object Detection

1. 在自上向下特征融合之后，再自下而上进行融合
2. 在融合特征时，由于不同的输入特征具有不同的分辨率，它们通常对输出特征的贡献不同。为每个输入增加了一个额外的权重。



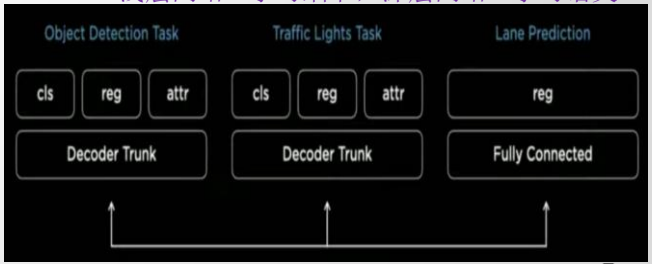
浅层网络—学习细节，深层网络—学习语义

head: HydraNet

2016 CVPR

You Only Look Once: Unified, Real-Time Object Detection

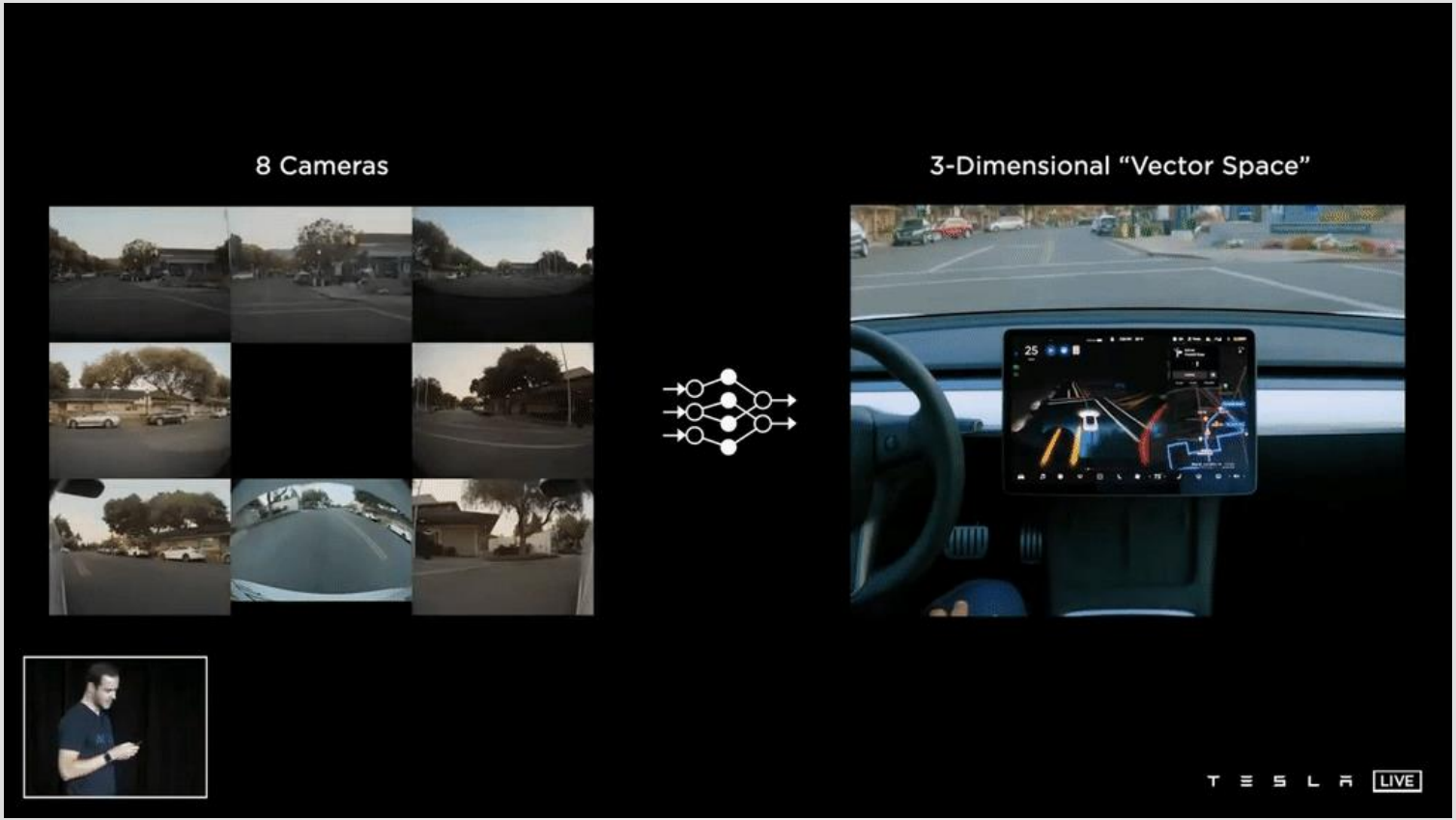
1. Feature Sharing : 减少重复卷积计算、backbone数量
2. De-Couples-Tasks：特定任务与主干解耦，能够单独微调任务
3. Representation Bottleneck：在训练过程中缓存特征，当在做微调工作流时，只使用缓存的特征对头部进行微调。



cls--分类，reg--回归边界框+中心

Input→backbone→neck→head→Output

Vector Space



感知系统预测的结果必须转化为三维空间，这是Plan & Control系统的基础。特斯拉将这个 3D 空间称为“**向量空间**”。车辆及其所在空间的信息，如车辆的位置、速度、车道、标志、信号灯和周围物体等。

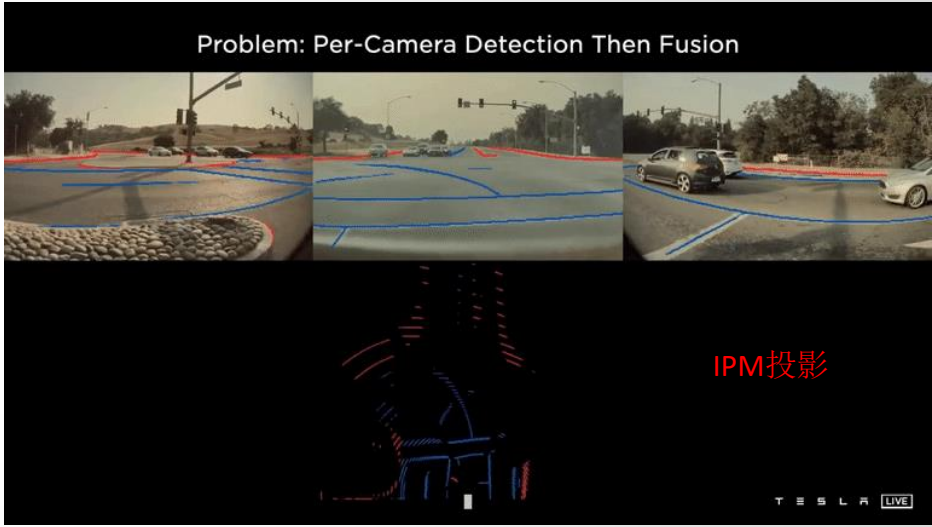
Vector Space

Problem 1 - Occupancy Tracker

Border or Freespace：

1. 单相机内部目标的 tracker 要处理各种 case，很难， 相机之间更是如此。
2. 目前单目的测距模型依赖了非常多的先验条件（大概率不能完全满足），造成投影误差根本没办法控制。

Problem 2 - Per-cam detection then fusion



Lane：

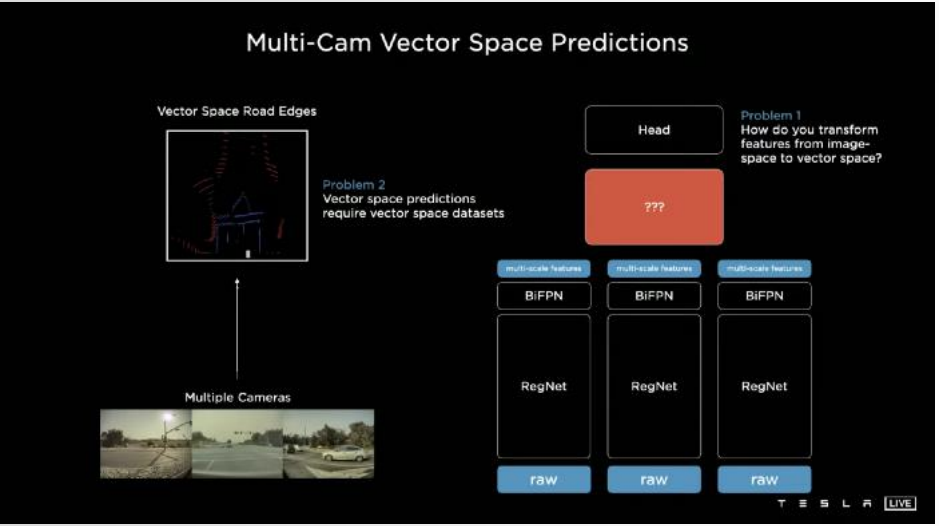
1. 图像每像素对应的物理空间，随着距离增加会越来越大，也就是解析度会降低。
2. 一旦先验条件不满足度增加，整个投影结果都会出现不稳定。

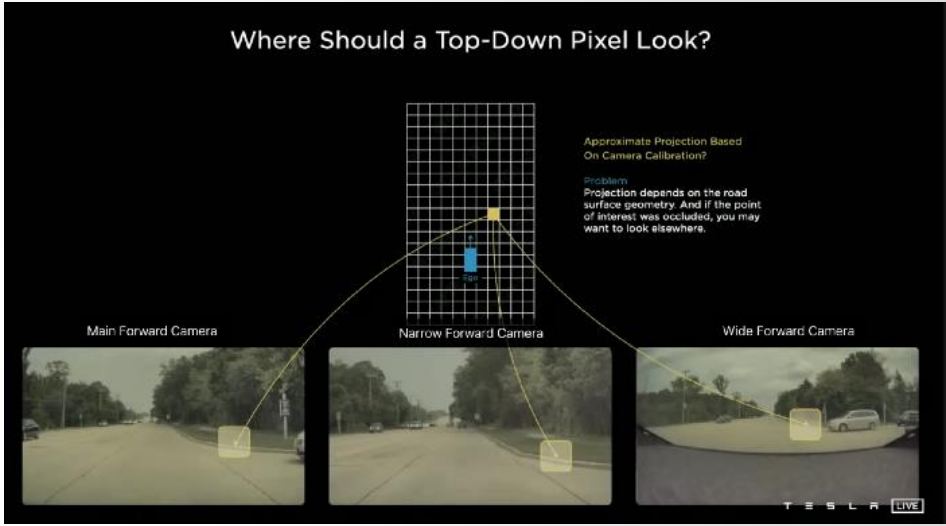
Detection：

1. prediction of occluded areas
2. prediction of larger objects (an object spans multi cameras)

Multi-Cam Fusion

We'd like to take **all of the images** and simultaneously feed them into **a single neural net** and directly output in **vector space**.





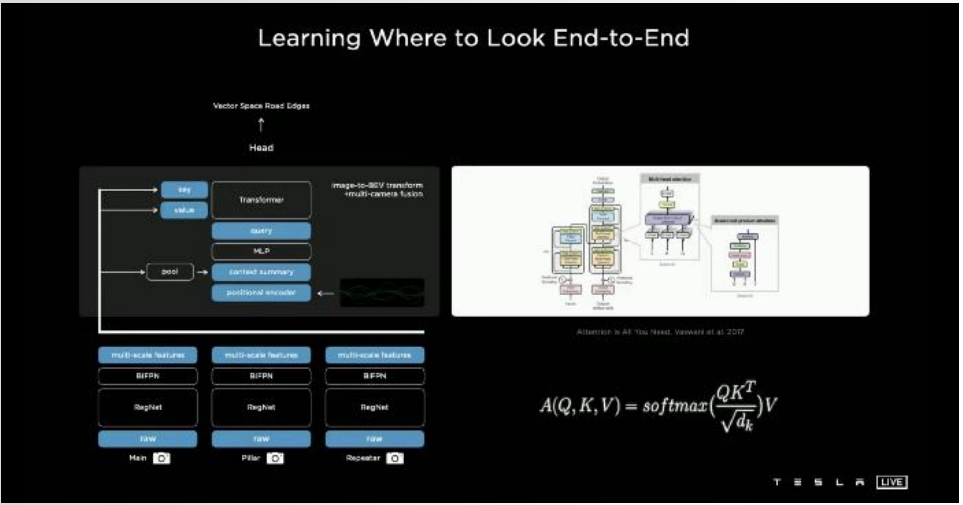
Two difficulties：

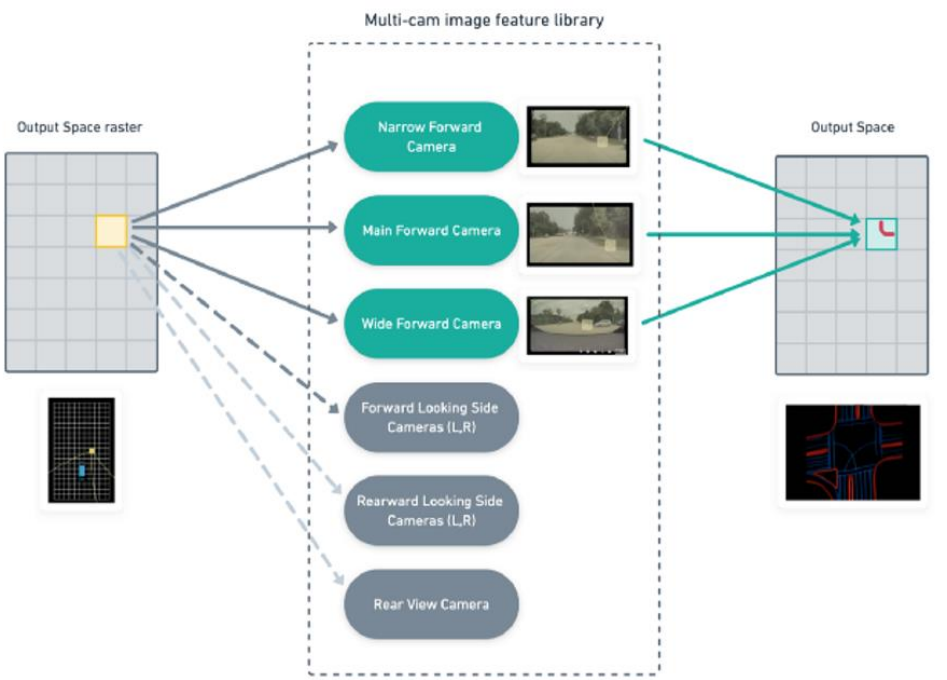
1. 如何将特征从**图像空间转换到向量空间**？ 以及如何使其可区分，以便端到端训练成为可能。
2. 所有的优化方法只有在函数可微时才有效。如果想在神经网络中进行向量空间预测，你需要基于向量空间**特定的数据集**。

Two technologies：

1. Learning where to look.(image->BEV + multi-cam fusion)
   multi-cam, multi-scale features -> Transformer -> BEV
2. Variations in Camera Calibration(Rectify)
   Camera Calibration -> virtual common camera

Transformer





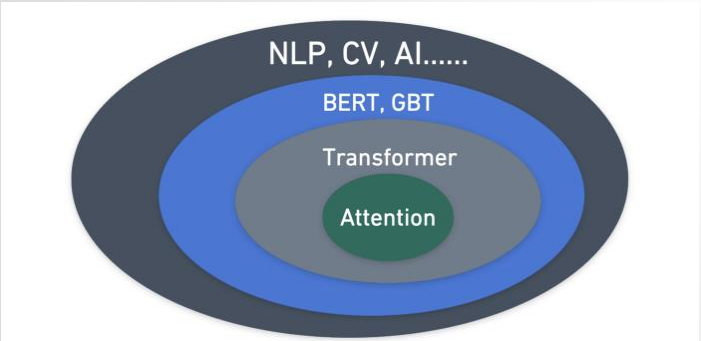
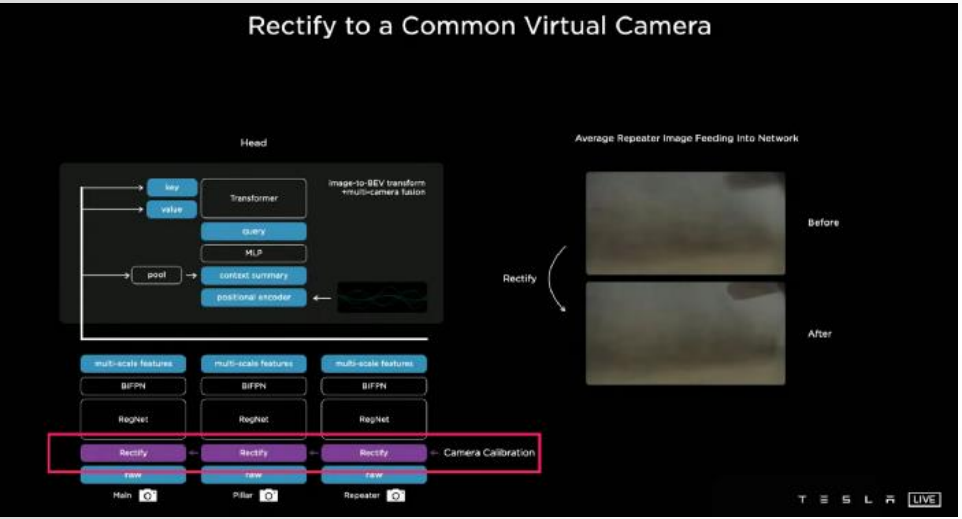


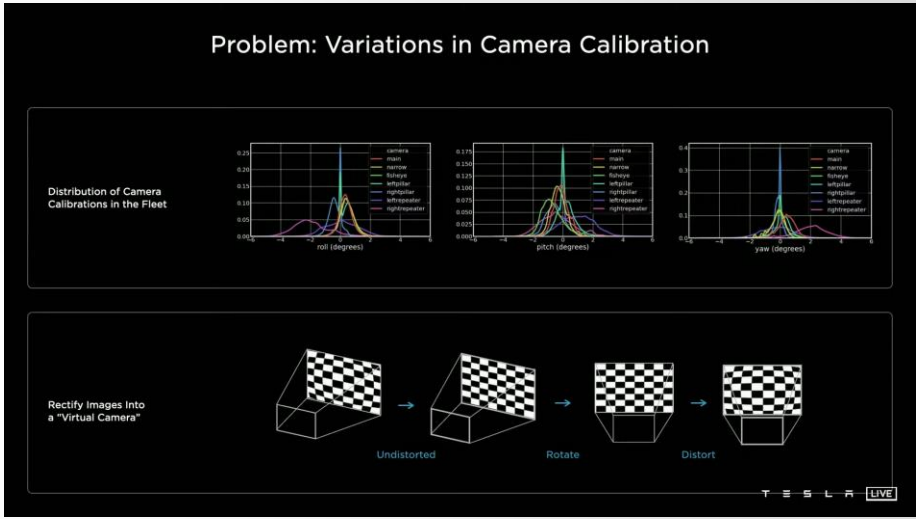
Image Space转换为Vector Space的过程：

1. 初始化一个输出空间大小的栅格：Output Space Raster
2. 对输出空间栅格上的点进行位置编码。接下来，使用多层感知器（MLP）将其编码为一组查询向量，例如黄色栅格。
3. 所有图像（来自 8 个摄像头）及其特征也发出自己的键和值。（Multi-Cam image feature library）
4. keys和querys交互乘法（Transformer中的dot-product attention）在Multi-Cam图像特征库中搜索并将结果输出到向量空间。

Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//Advances in neural information processing systems. 2017: 5998-6008.

Virtual Camera



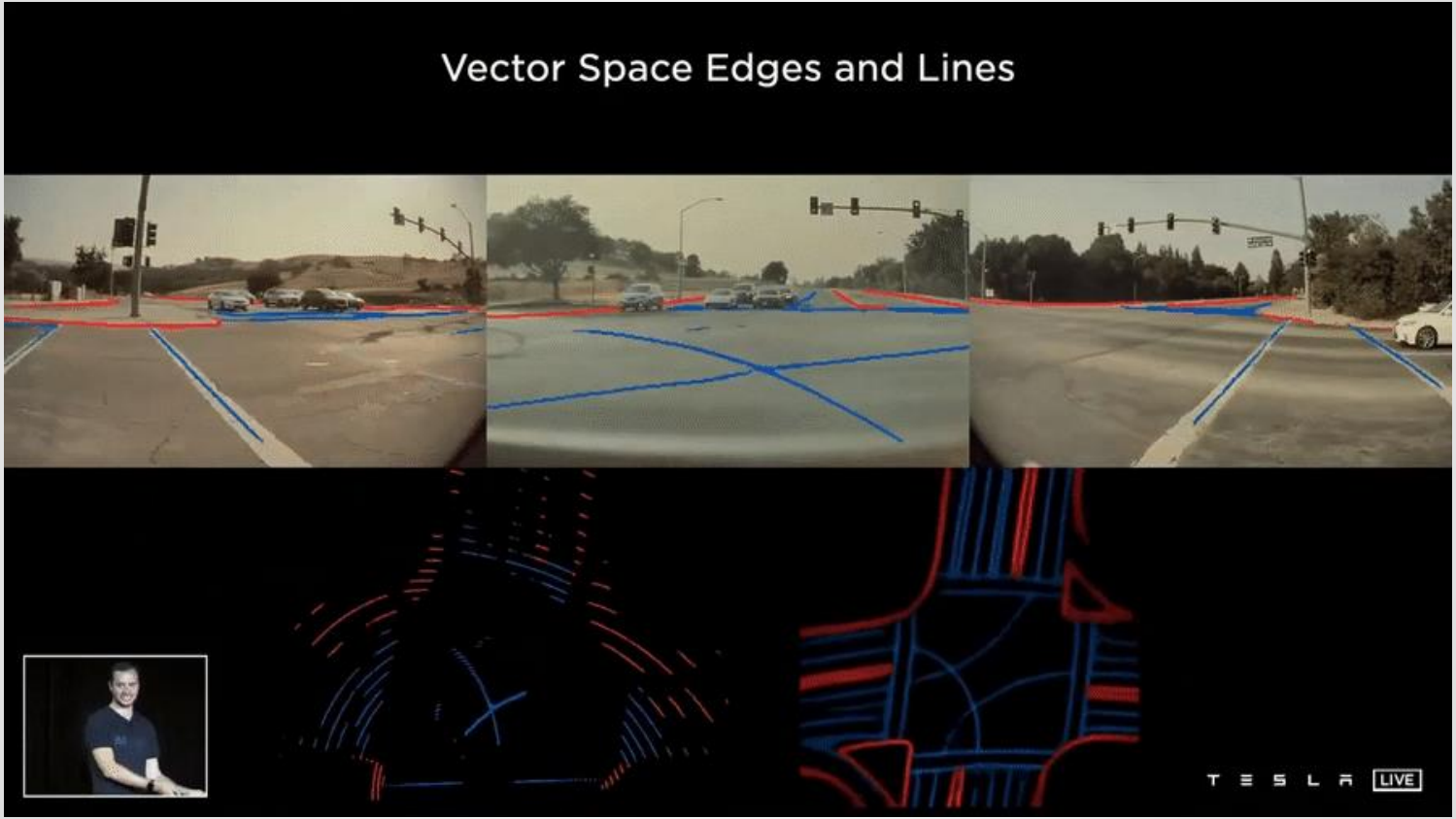


1 . 由于通过模型前融合多相机数据，模型内部隐式的学习到了多相机的位置、视角等参数。但是在量产车中，这个多相机的空间关系是不能保证一致性的：每一辆车的相机组成都有一定的误差，如何消除这个误差？

2. Tesla 这里的做法是通过标定，重新调整所有的车辆的摄像头到一个标准视角上：**Camera Calibration -> virtual common camera**

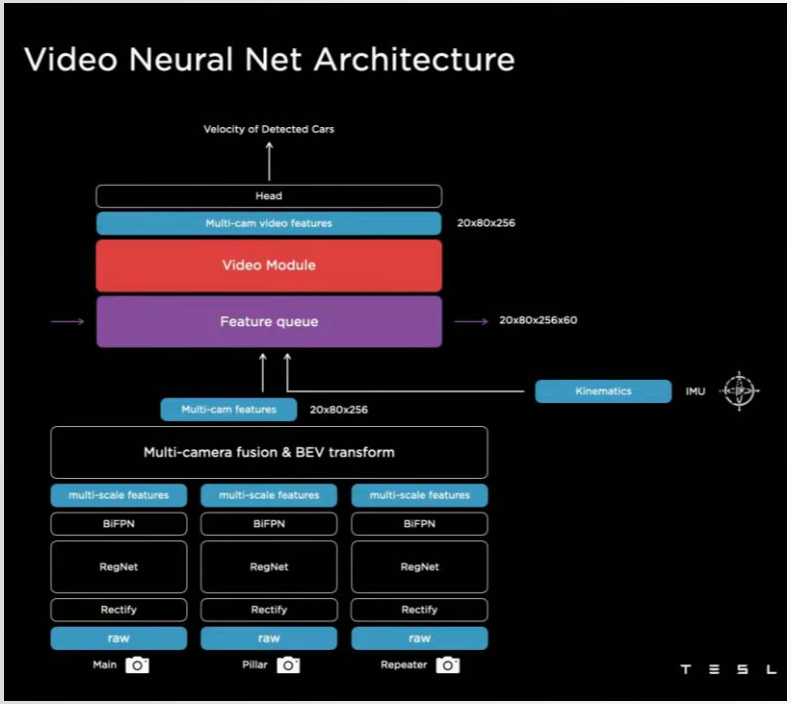
3. 思路就是，先标定好相机，对图像矫正去畸变、变换RT，最终都投射到一个标准的 virtual common camera 的视角上。这样就可以保证，量产车上的所有的摄像头空间关系都和训练时的视角、畸变一模一样。（归一化）

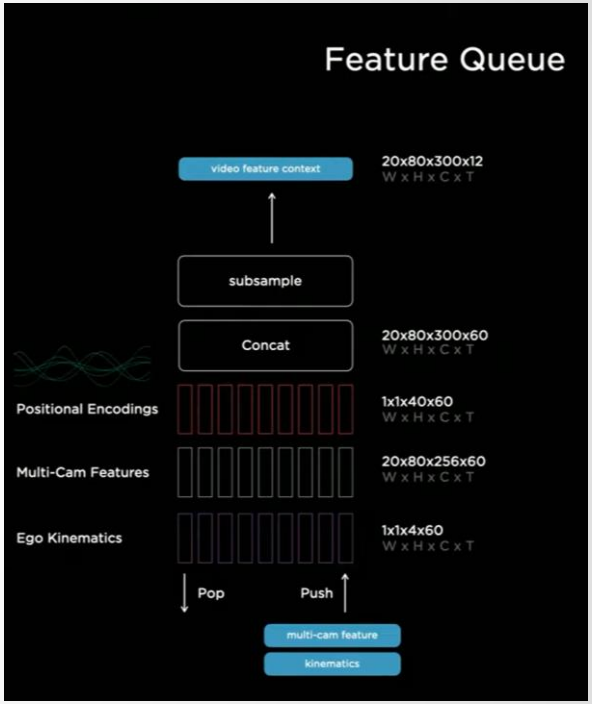
Vector Space



Left: Image Space detection + fusion Right:Vector Space + multi-camera fusion

Video Module （时序融合）



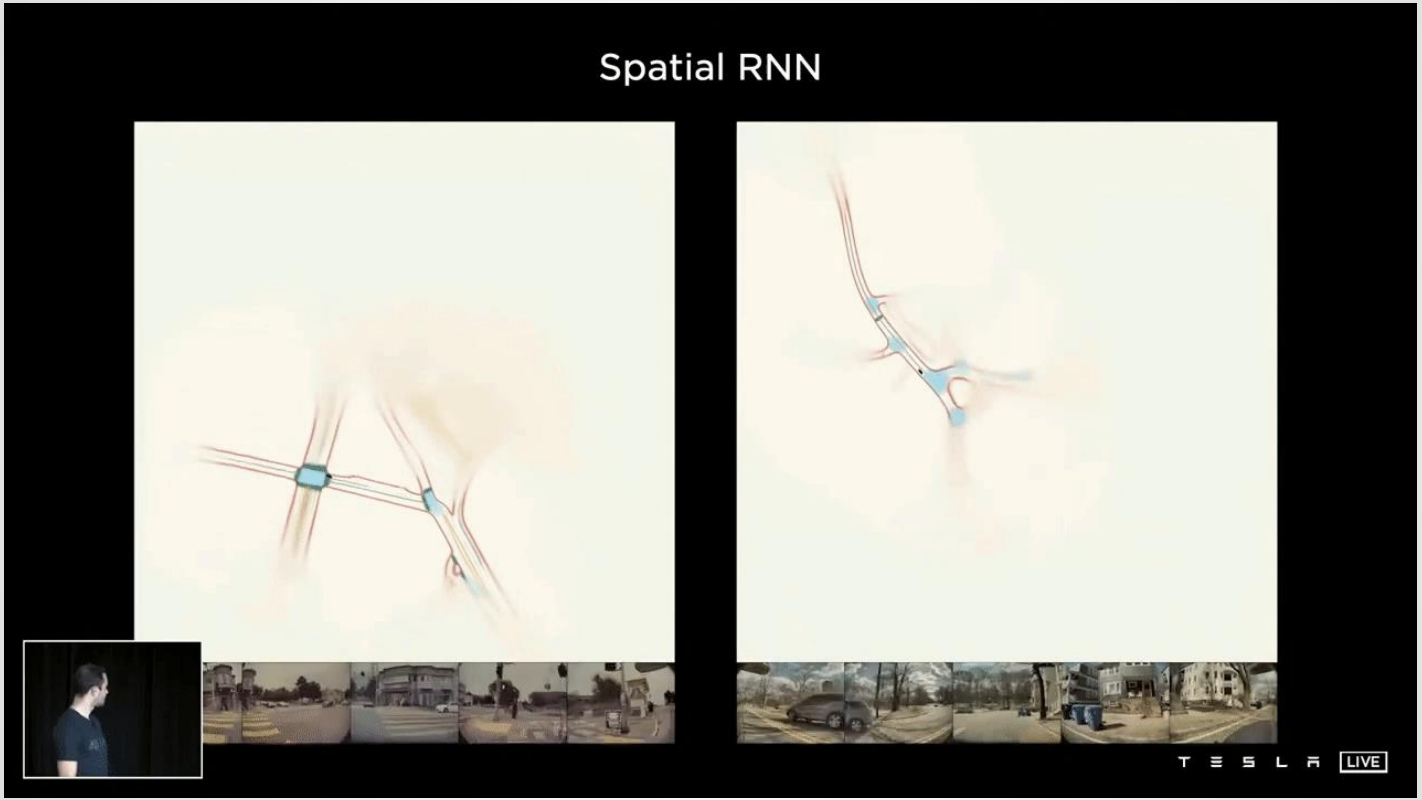


感知的都知道纯粹的单帧结果没有任何意义。很多状态无法识别，比如：速度/方向灯状态/目标的遮挡与重现

Sequence sample： Time-based: 27ms 周期
Space-based 1 m 周期

融合模块： 3d conv/Transformer/RNN 都试了，RNN的效率最高，重点介绍的是 Spatial RNN

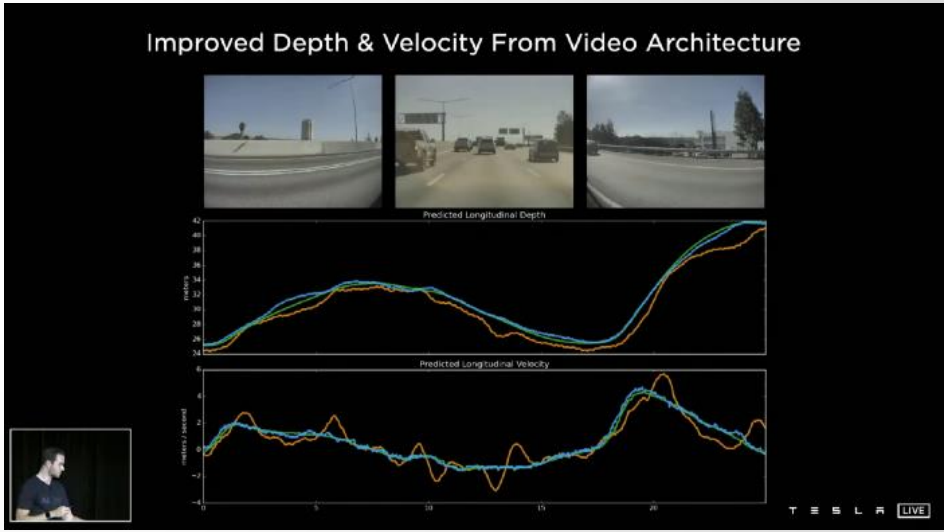
Spatial RNN



1. 先建立一个 BEV 的大图，譬如 1000x1000，车辆的感知范围可能是 30x40。
2. BEV 视角下的每一个像素点（or pillar，如果搞激光的话）都用一个 RNN 来维持状态。
3. 只要感知范围覆盖了，就用对应特征跑一步RNN。

Spatial RNN

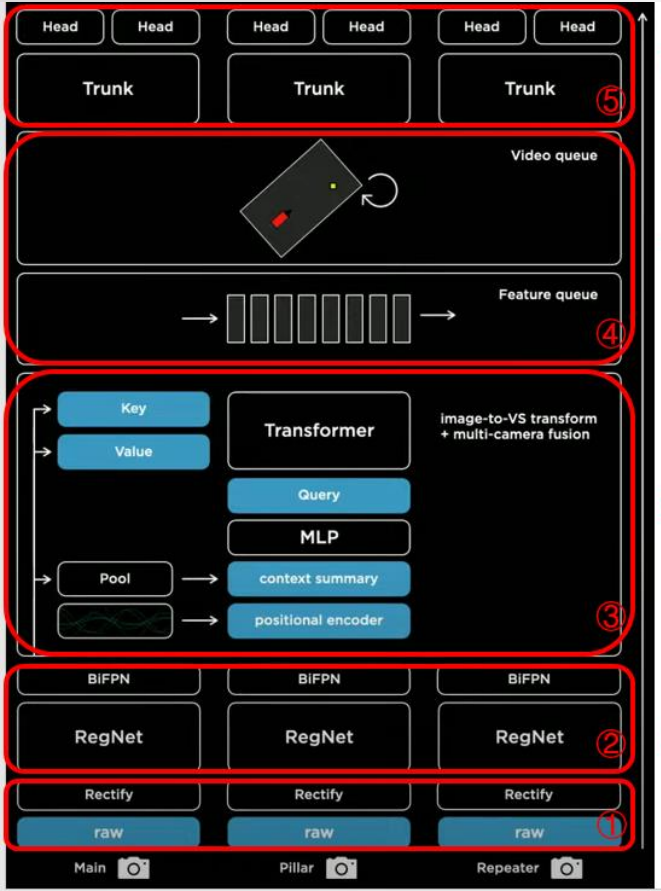
The Benefits of Spatial RNN:



1. Improved Robustness to Temporary Occlusion

2. Improved Depth & Velocity From Video Architecture

Tesla Vision Network Final Structure



HydrNet for Multi-Tasks

Time-Based & Space-Based Memory

Multi-Cam Fusion + Vector Space

Sigle-Cam Object Detection

①. **Raw images** feeding on the bottom and go through **rectification** layer to correct for camera calibration and put everything **into a common virtual camera**,

②. Pass them through **RegNets** residual networks to process them into a number of features at different scales and fuse the multi-scale information with **BiFPN**

③. Goes through a **transformer** module to re-represent it into the **vector space**, and output space.

④. Feeds into a featured queue in time or space that gets processed by a **video module**, like the **Spatial RNN**.

⑤. The continues into the branching structure of the **HydraNet** with trunks and heads for all the **different tasks**.

03

Others

Others

特斯拉下一步计划：

1、 当前网络结构融合的位置过于靠后端，后续特斯拉考虑引入光流等信息进行网络结构上早期信息的融合；

2、 当前网络输出结果仍然是密集的（dense），这将加重后续处理负担，造成不理想的时延，后续将探索更加稀疏（sparse）的数据结构表达输出结果，提升实时性。

关注点：

- 主力 **Vision**：依靠 Lidar + HDMap 的方案成本较高、通用性不足、长期维护问题难以解决。
- **8相机**保持36Hz 的速度。从技术角度来看，低像素高帧率的相机是更加务实的选择。如果上好几个 4K 分辨率，算力、带宽都是瓶颈。
- **数据**要数量多、质量好、覆盖长尾。当然，这其实是一句废话，大家都知道。只不过特斯拉是踏踏实实在用这个标准采数据。
- 提了下 **Auto Labeling**，基本思路是：依靠专家模型，更加复杂的离线算法，多传感器，前后时序关系，***还有人工验证与调整***。
- 数据收集这块，依靠 200+ 的 **trigger**，slides里面举了例子（16:00的位置上），算是利用车上传感器、司机，各种脑洞。
- Data Engine这块，业界公认思路：不断的进行数据迭代。特斯拉真的有几百万量车在路上跑着，利用上面的那些 trigger 采集数据。
- **网络结构**这块不罗嗦了，一般反而是DL外行喜欢第一时间找网络结构看一下，似乎结构才是解决问题的关键（然而，并不是）。
- **算力平台**就是凸显我很牛逼、算力强，没有介绍细节，应该也不是 Andrej 本人的强项。不过实际上，超算里面带宽、线路设计、硬件配置如果要用满性能都是要经验的。
- **FSD computer** 几句话着重的都是部署操作：算子合并、int8、编译器。比较常规，也没说 QAT 有没有精度衰减。
- **技术栈要全打通**：Tesla 自己造车、自建超算、自己搞车载平台、自己研发、自己的（用户）车队采集，这样的整合能力是其他任何厂商目前都不具备的。

Tesla AI Day

https://www.youtube.com/watch?v=j0z4FweCy4M

Andrej Karpathy (Tesla): CVPR 2021 Workshop on Autonomous Vehicles

https://www.youtube.com/watch?v=NSDTZQdo6H8

04

Q & A

Thanks