MMDetection是一个目标检测工具包，有许多目标检测方法和实例分割方法。它不仅包括训练和推理的代码，也提供超过200个网络模型的权重文件。通过MMDetection，我们不仅能复现已有的方法，而且还能开发自己的新的检测器。

**一、MMDetection1.0 主要特点：**

1. 模块化的设计：检测框架被分解成多个不同的模块，通过组合不同模块可以简单构建墓边检测框架。
2. 支持多种框架开箱即用
3. 高效：所有基础的bbox，mask在GPUs上进行运算，训练速度比detectron，maskrcnn-benchmark，SimpleDet快。

本文的结构：MMDetection中支持的模型和框架；基于benchmark的模型实验结果；展示在一些基线上的消融研究

**二、支持的框架：**

**单阶段检测方法**：SSD，RetinaNet，GHM，FCOS，FSAF

**两阶段检测方法**：Fast R-CNN，Faster R-CNN，R-FCN，Mask R-CNN，Grid R-CNN，Mask Scoring R-CNN，Double-Head R-CNN

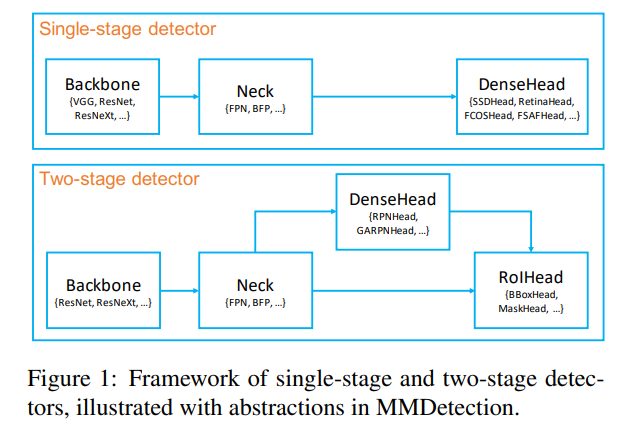
**多阶段检测方法**：Cascade R-CNN，Hybrid Task Cascade

**通用模块和方法**：Mixed Precision Training，Soft NMS，OHEM，DCN，DCNv2，Train from Scratch，ScratchDet，M2Det，GCNet，Generalized Attention，SyncBN，Group Normalization，

Weight Standardization，HRNet，Guided Anchoring，Libra R-CNN

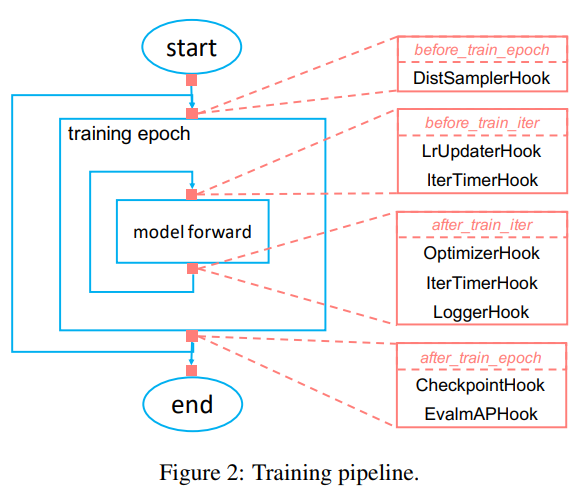
**三、架构：**

**模型展现**：**Backbone**（ResNet-50）将image转化成feature maps，**Neck**（FPN）对raw feaure maps进行提纯，**DenseHead**（AnchorHead，AnchorFreeHead：比如RPN-Head，Retina Head，FCOSHead），**RoIExtractor**（RoIPooling），**RoIHead**（BBoxHead/MaskHead，将RoI作为输入，并对RoI进行预测，比如边界框分类/回归，mask预测）



**训练流水线**：使用hooking机制进行统一训练流水线的设计，这个训练流水线可以用来进行目标检测，图片分类和语义分割。很多任务训练过程共享相似的工作流：training epochs，validation epochs（可选），为了实现通用训练，需要定义一些运算来处理某个特定的Steps

MMDetection定义了一些时间点，用户可以注册这些可执行的方法(hooks)，包括before\_run, before\_train\_epoch, after\_train\_epoch, before\_train\_iter, after\_train\_iter, before\_val\_epoch, after\_val\_epoch, before\_val\_iter, after\_val\_iter, after\_run.



**四、基准：**

**实验环境设置：**

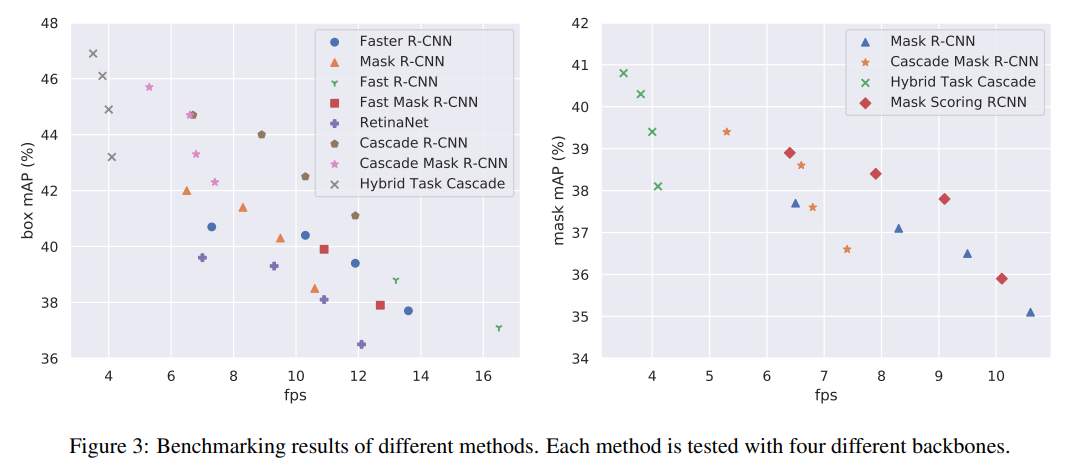
**数据集**：支持VOC，MS-COCO 2017 样式的数据集

**实现细节**：图片会resized至最大尺寸1333 x 800；使用8个V100 GPUs进行训练，使用单V100 GPU进行推理。

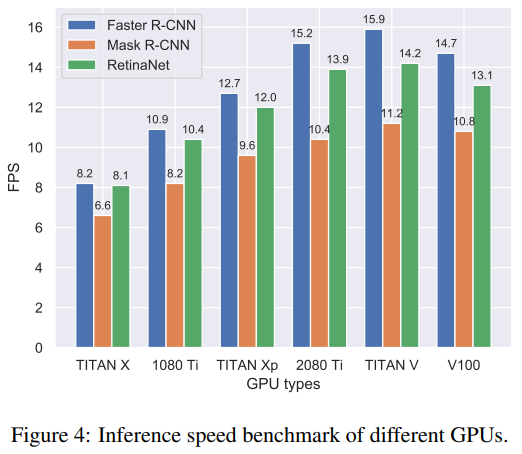
**评估指标**：**IoU**（0.5 ~0.95）用来评估COCO数据集，用AR，mAP来度量RPN网络

**基准结果（COCO）：**

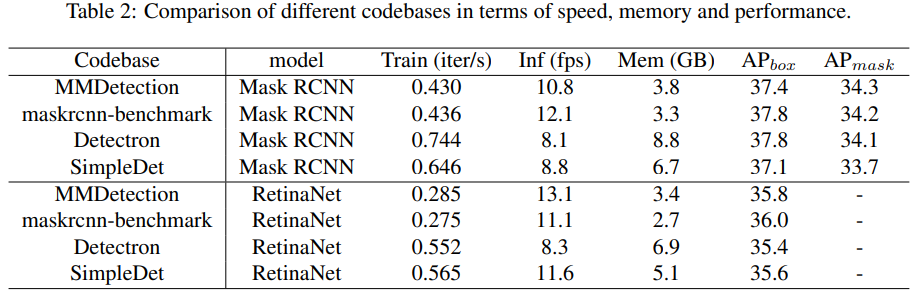
训练：使用4个backbones：ResNet-50, ResNet-101, ResNet-101-32x4d 和 ResNeXt101-64x4d（越大越好）



推理：

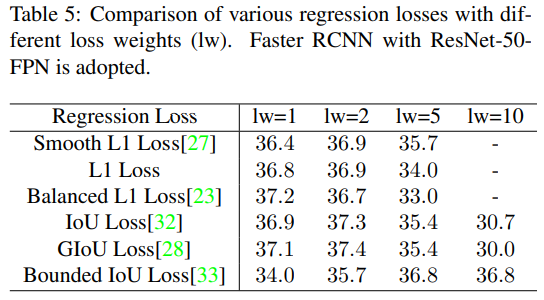


不同toolbox同个模型在训练和推理上的比较：发现MMDetection训练速度慢，但是消耗资源少。



**五、回归损失**

常用的回归损失：Bounded IoU Loss , IoU Loss , GIoU Loss, Balanced L1 Loss. L1 Loss。但是这些损失函数大部分已在不同的方法上实现或在设置上默认配置好了。



**六、归一化层**

Pytorch的细节部分…