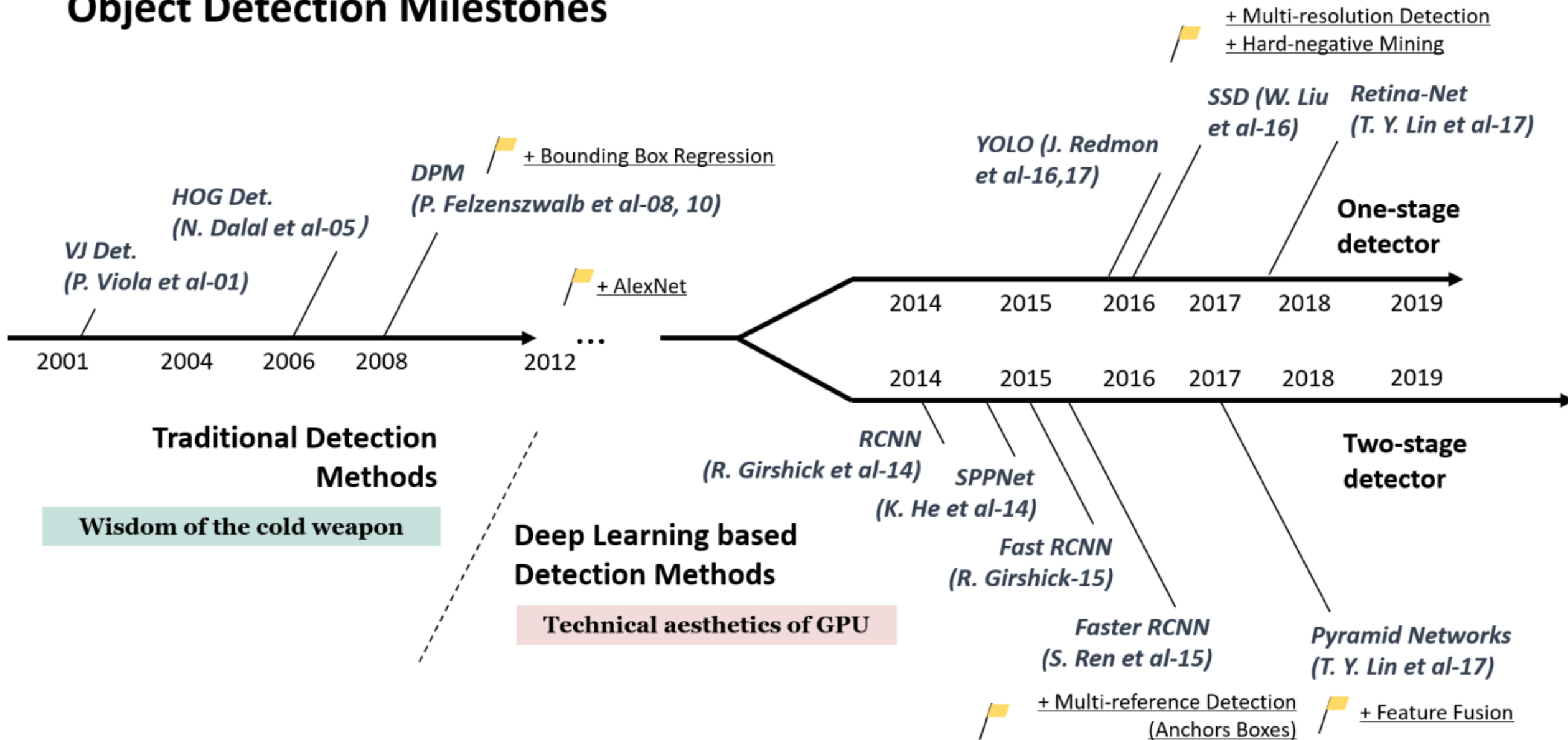
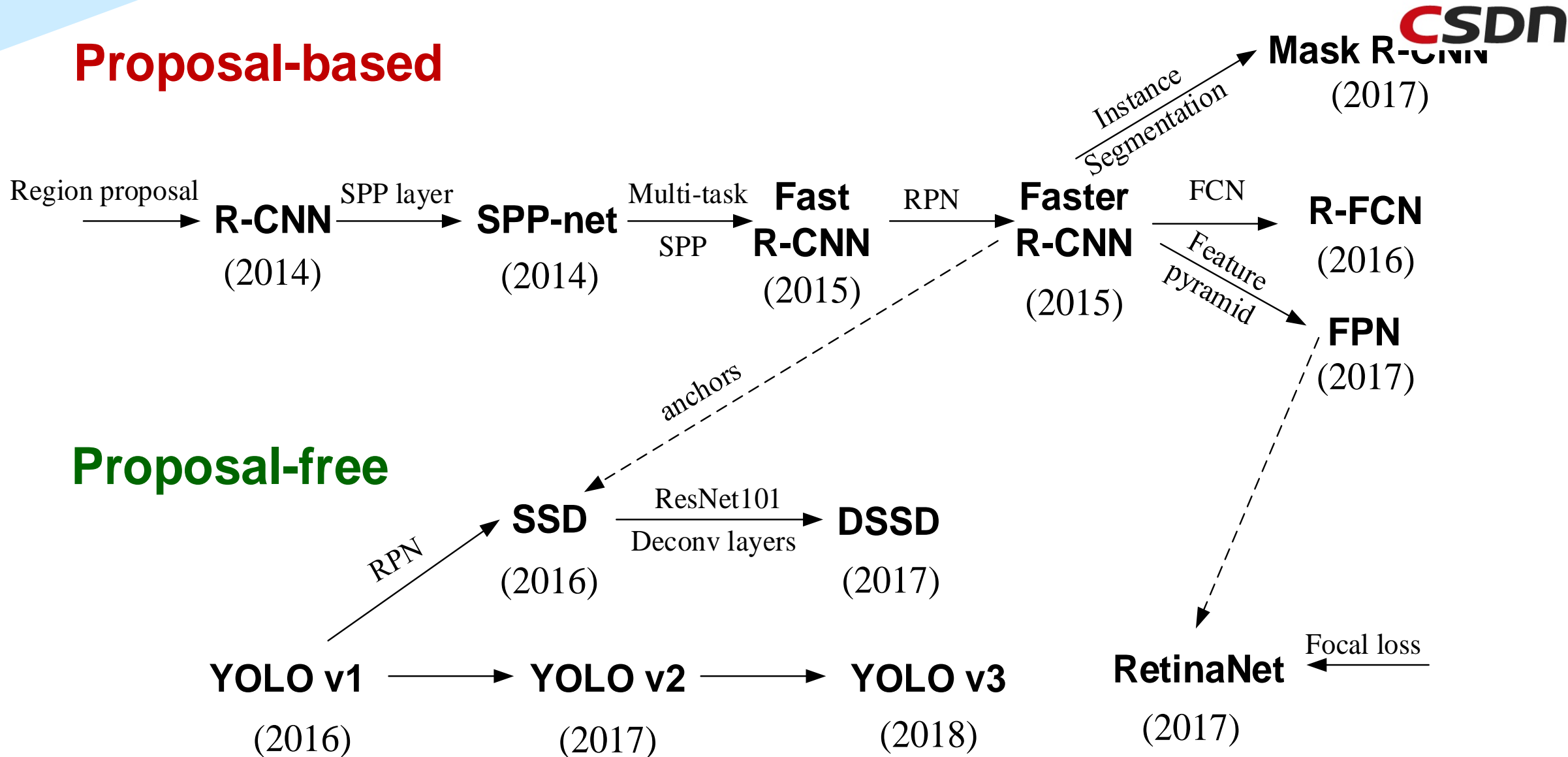


YOLO目标检测系列发展史

Object Detection Milestones



Proposal-based



Proposal-free

PyTorch版



Darknet

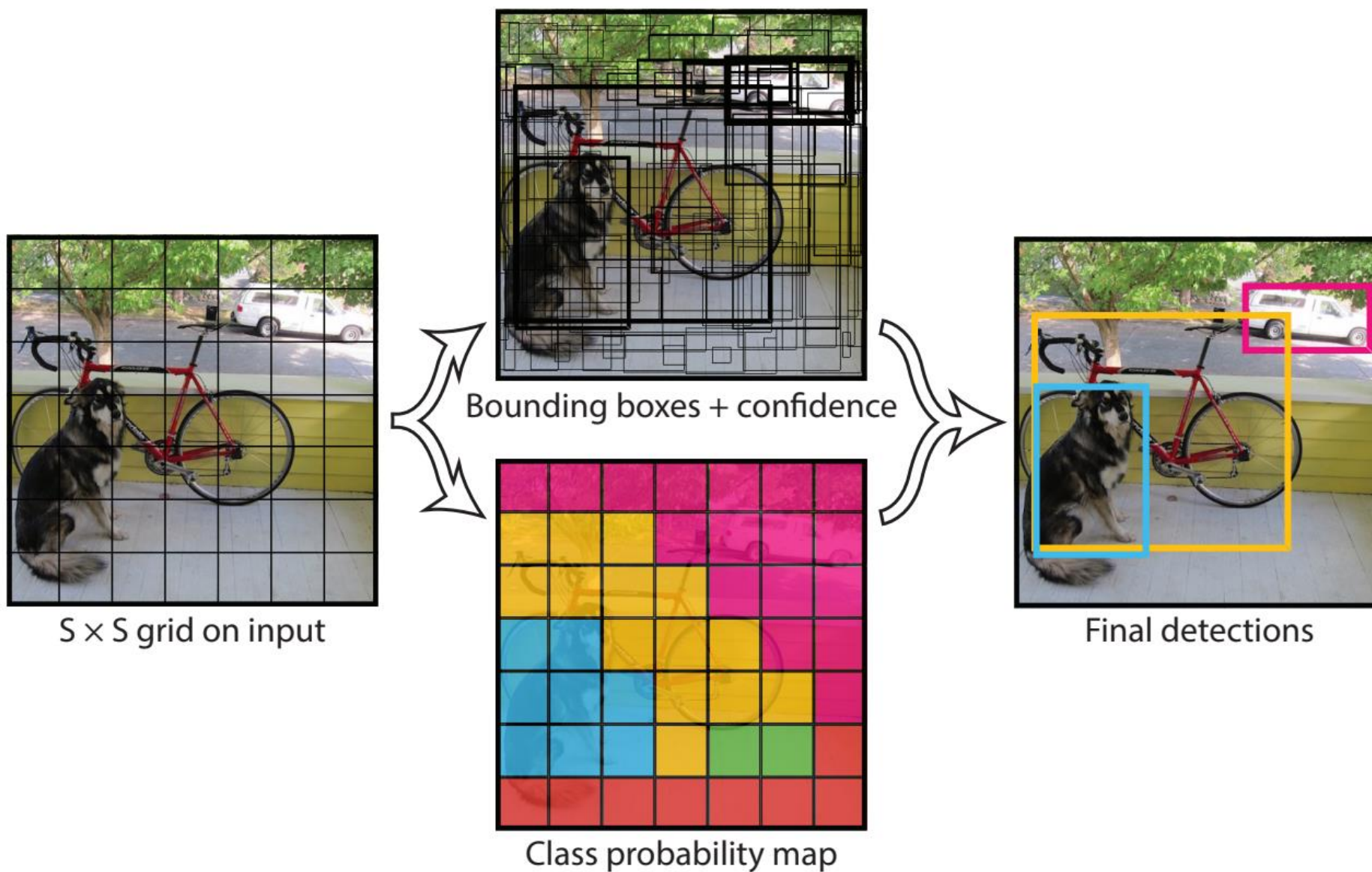
Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.

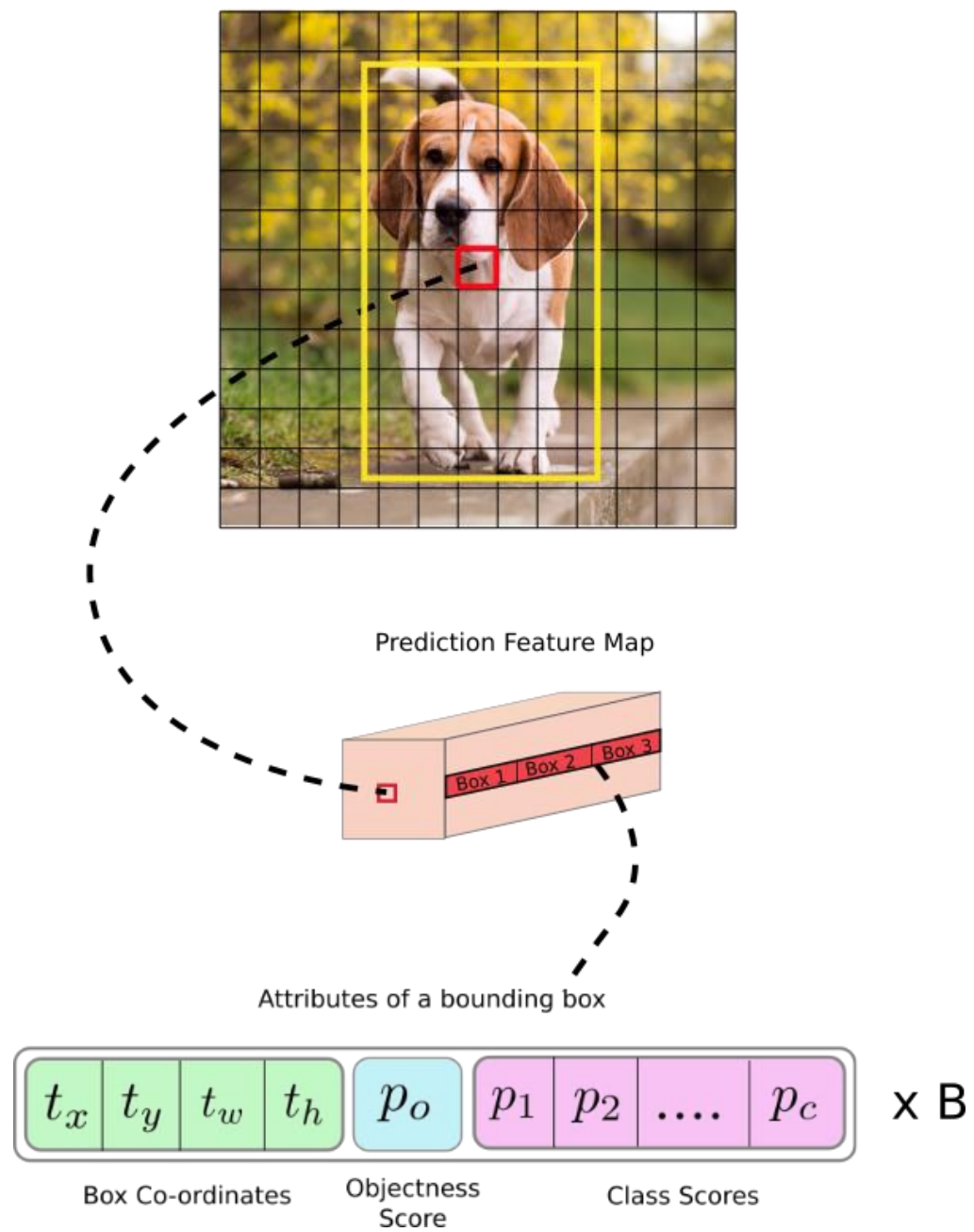
Yolo v4 paper: <https://arxiv.org/abs/2004.10934>

Yolo v4 source code: <https://github.com/AlexeyAB/darknet>

For more information see the [Darknet project website](#).

YOLO算法的基本思想

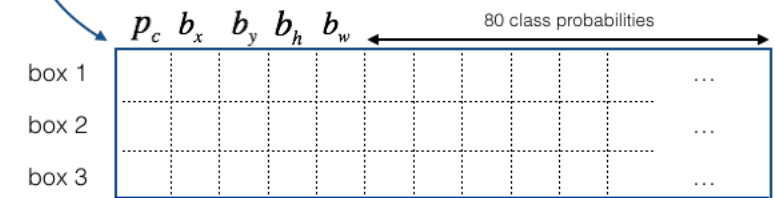
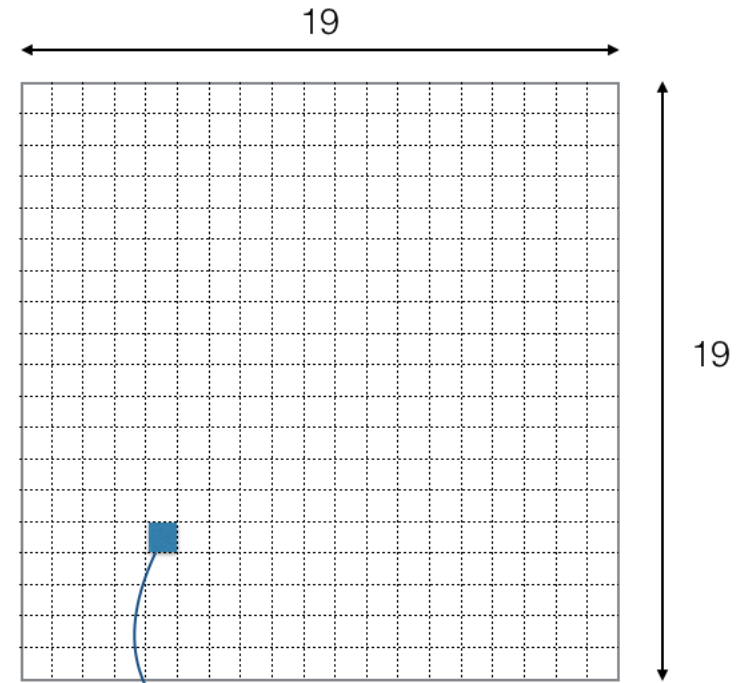




preprocessed image
(608, 608, 3)



Deep CNN
reduction
factor: 32

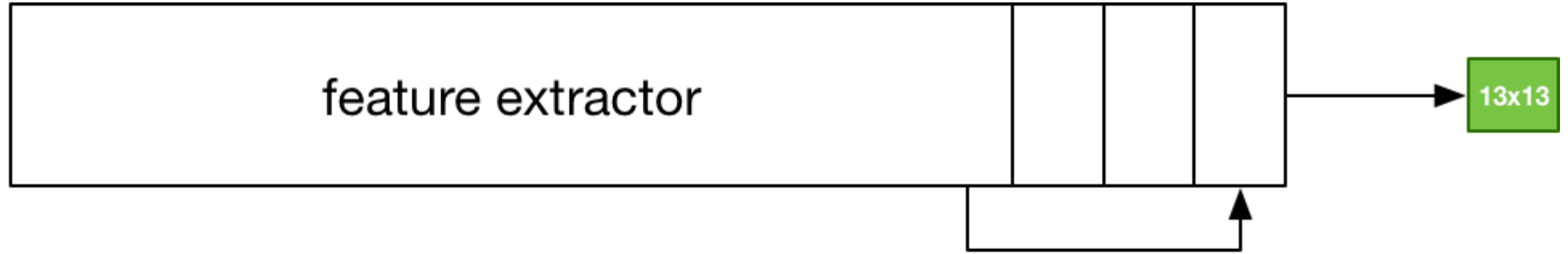
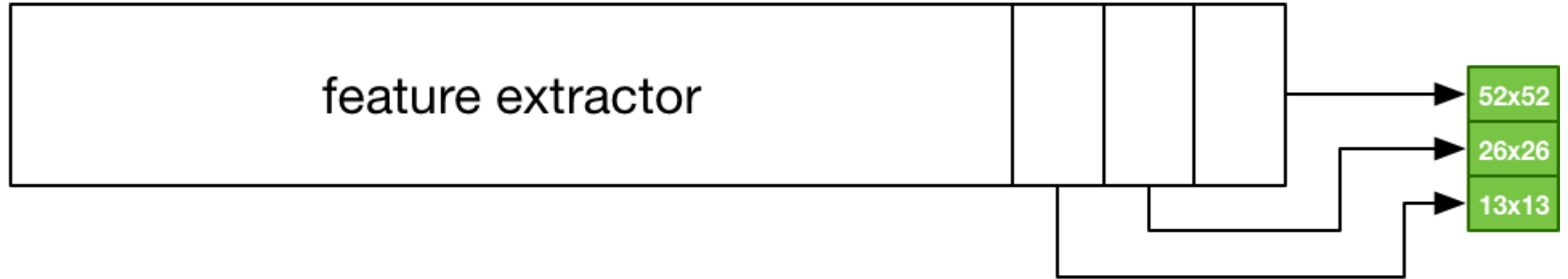


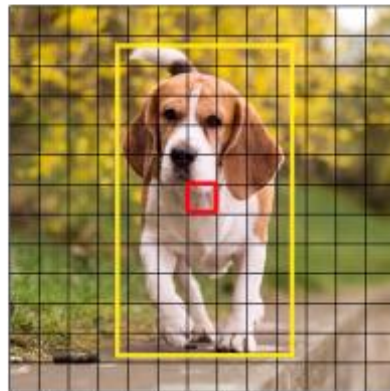
YOLOv3/v4算法的基本思想

- 首先通过特征提取网络对输入图像提取特征，得到一定大小的特征图，比如 19×19 （相当于 608×608 图片大小），然后将输入图像分成 19×19 个grid cells，如果GT中某个目标的中心坐标落在哪个grid cell中，那么就由该grid cell来预测该目标。
- 预测得到的输出特征图有两个维度是提取到的特征的维度，比如 19×19 ，还有一个维度（深度）是 $B \times (5+C)$ 。其中B表示每个grid cell预测的边界框的数量（YOLO v3/v4中是3个，即 $B=3$ ）；C表示边界框的类别数（没有背景类，所以对于VOC数据集是20）；5表示4个坐标信息和一个目标性得分（objectness score）。

类别预测(Class Prediction)

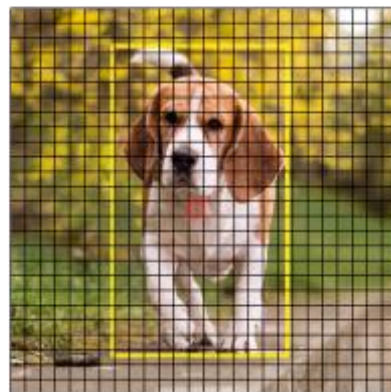
- 大多数分类器假设输出标签是互斥的。如果输出是互斥的目标类别，则确实如此。因此，YOLO应用softmax函数将得分转换为总和为1的概率。而YOLOv3/v4使用多标签分类。例如，输出标签可以是“行人”和“儿童”，它们不是非排他性的。（现在输出的总和可以大于1）
- YOLOv3/v4用多个独立的逻辑（logistic）分类器替换softmax函数，以计算输入属于特定标签的可能性。在计算分类损失时，YOLOv3/v4对每个标签使用二元交叉熵损失。这也避免使用softmax函数而降低了计算复杂度。

YOLO v2**YOLO v3**

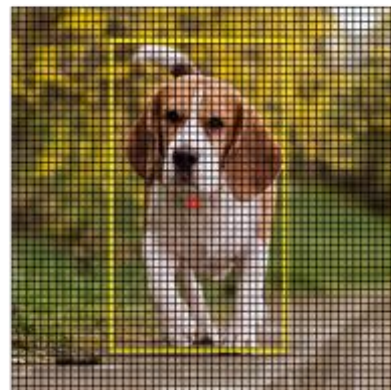


416×416网络分辨率

13 x 13

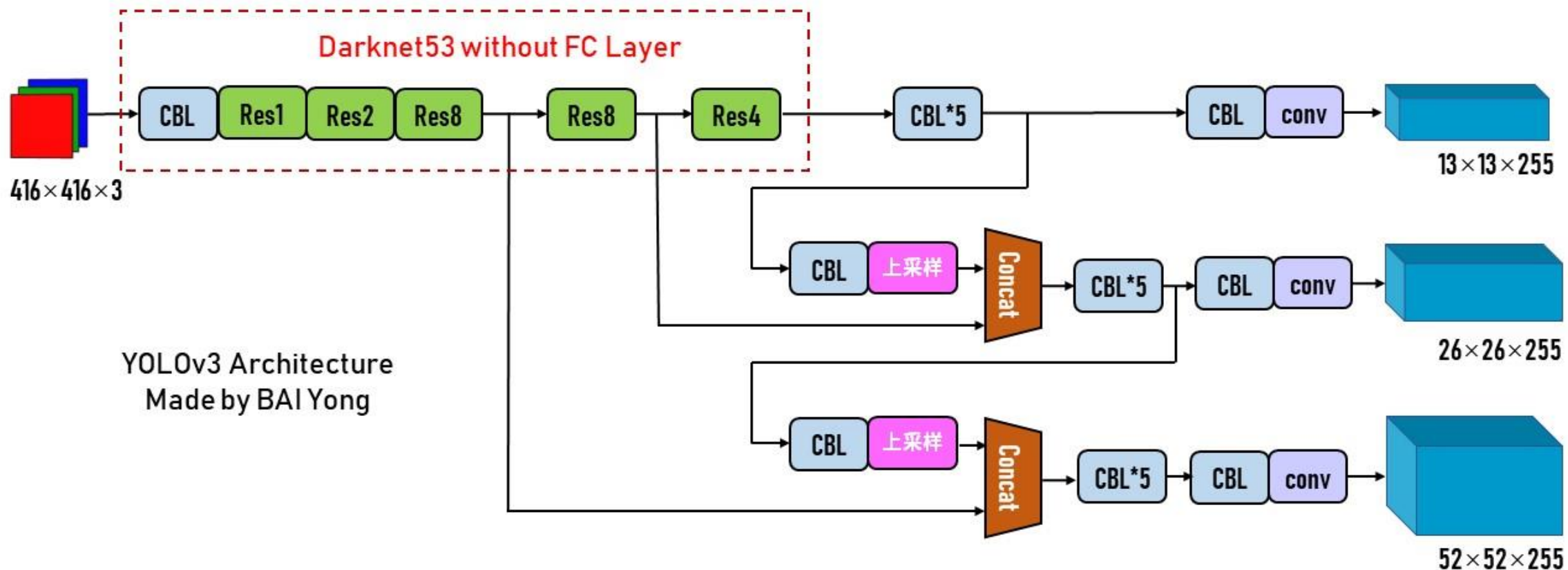


26 x 26



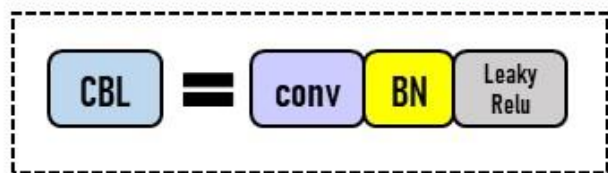
52 x 52

YOLOv3 网络架构

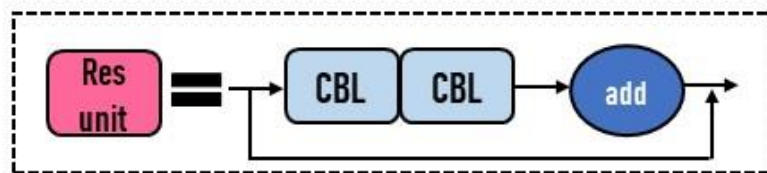


YOLOv3 Architecture
Made by BAI Yong

CBL (DarknetConv2D_BN_Leaky)



Res_unit



Resblock_body



YOLOv4 网络架构

