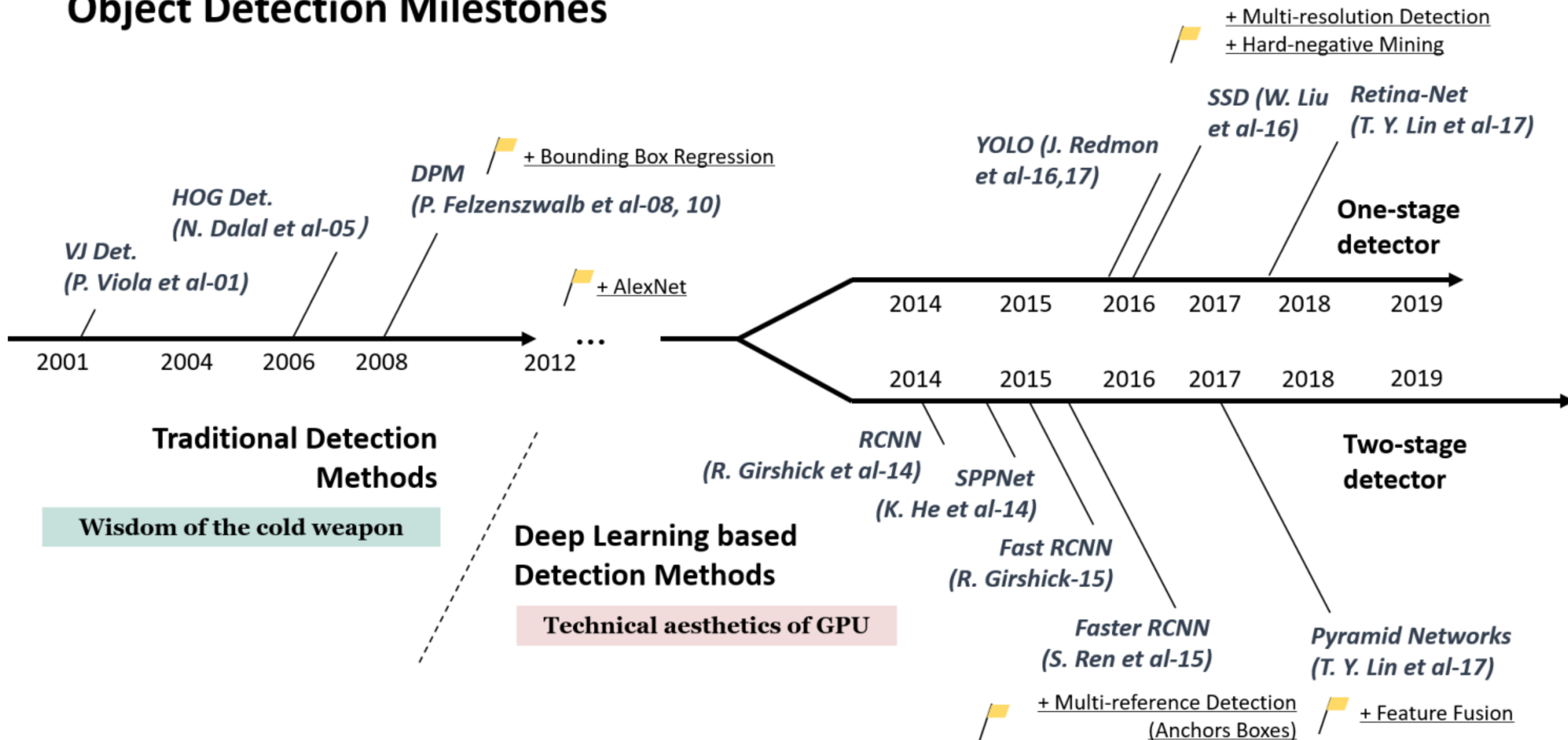
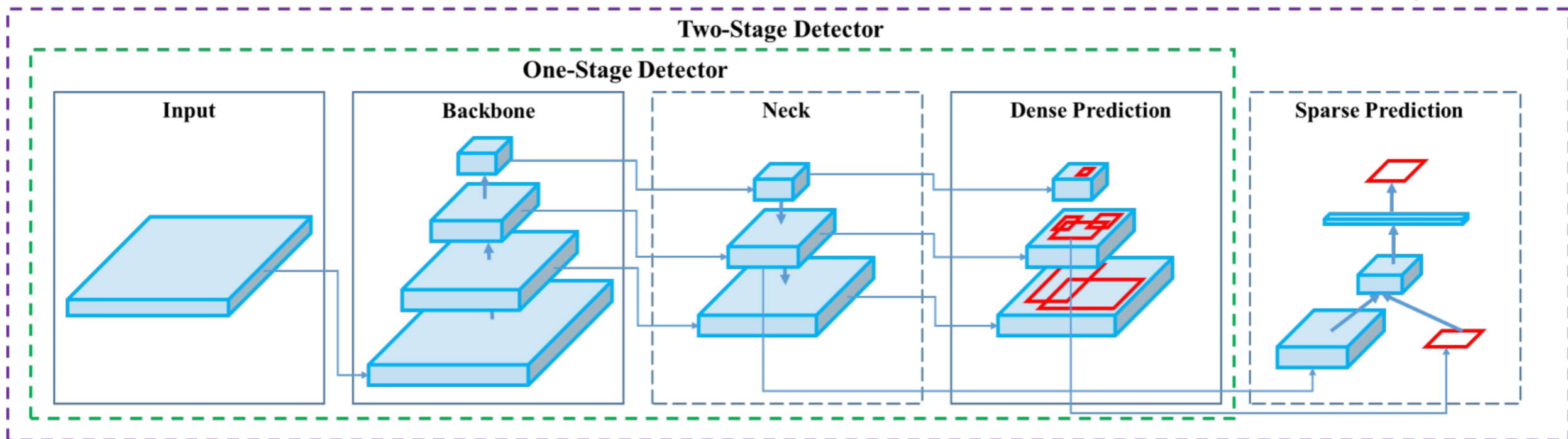


YOLO目标检测系列发展史

Object Detection Milestones





Input: { Image, Patches, Image Pyramid, ... }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], ... }

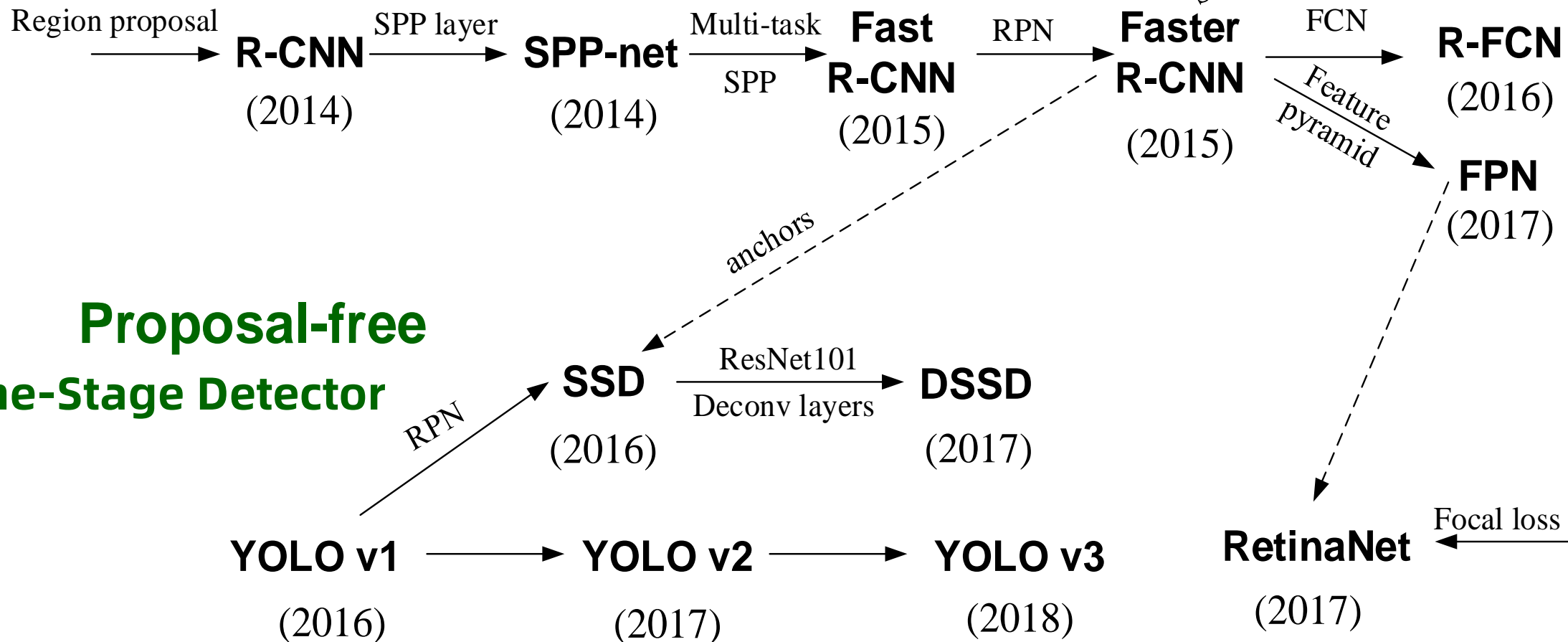
Neck: { FPN [44], PANet [49], Bi-FPN [77], ... }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], ... }

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], ... }

Proposal-based Two-Stage Detector



PyTorch版

YOLO v4 (2020.4)

YOLOv5 (2020.6)



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.

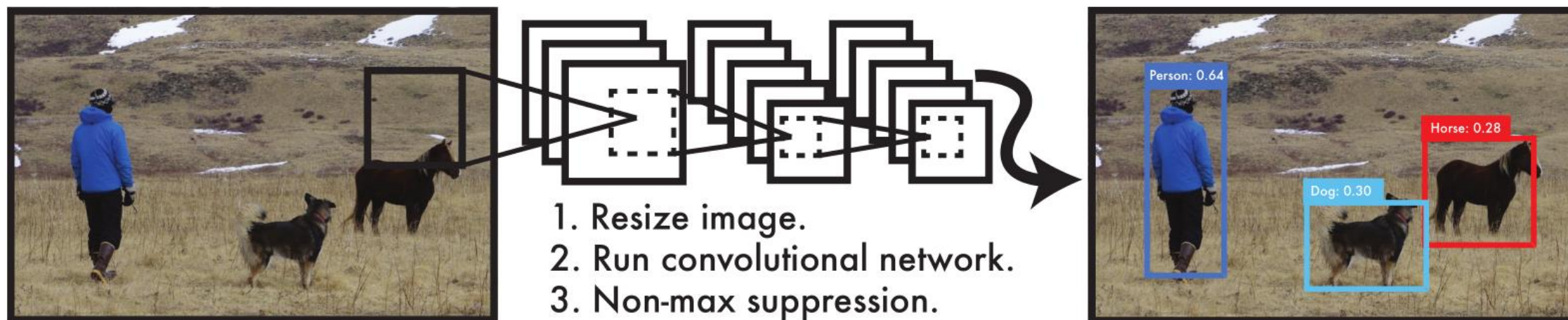
Yolo v4 paper: <https://arxiv.org/abs/2004.10934>

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

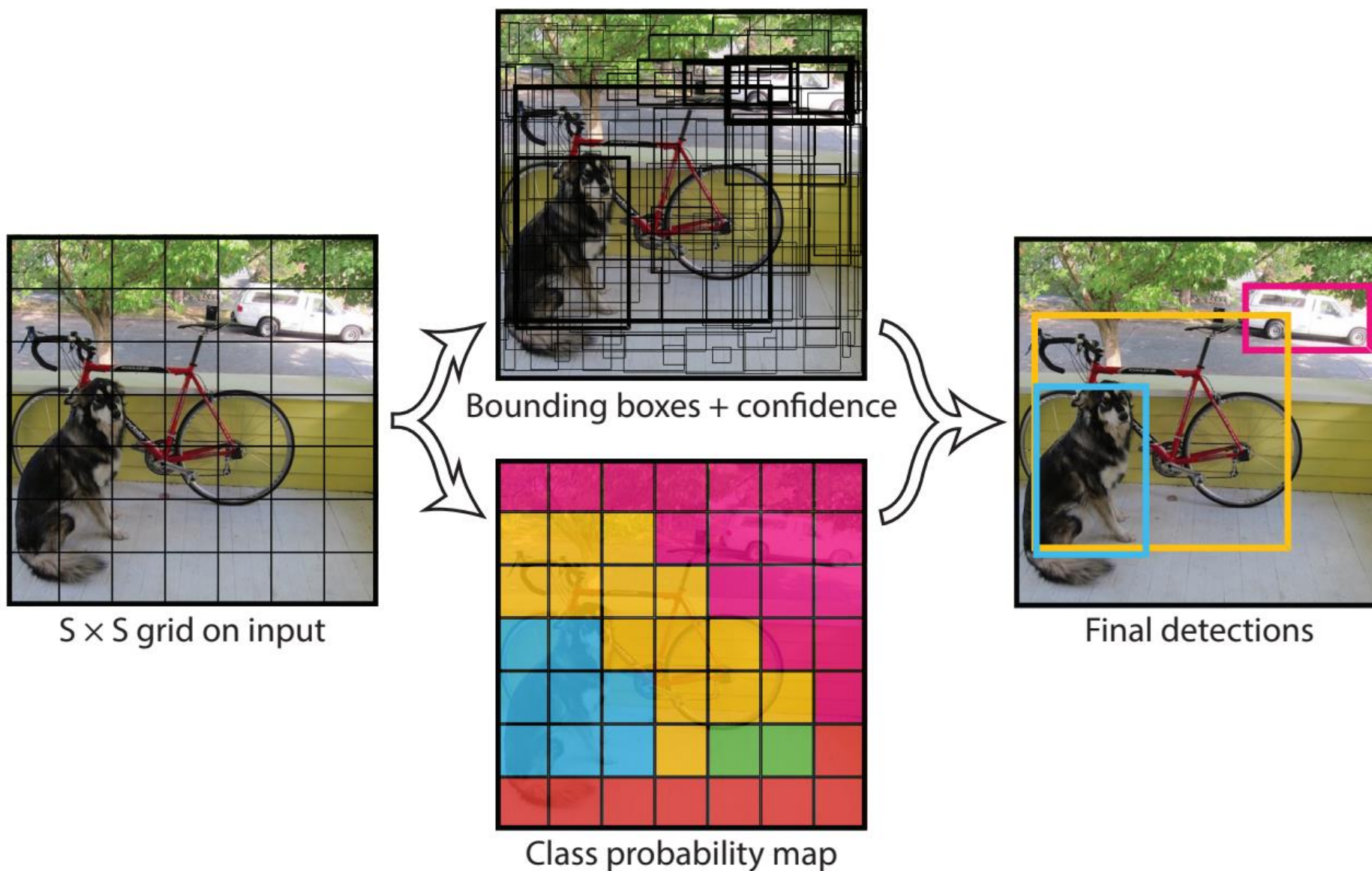
For more information see the [Darknet project website](#).

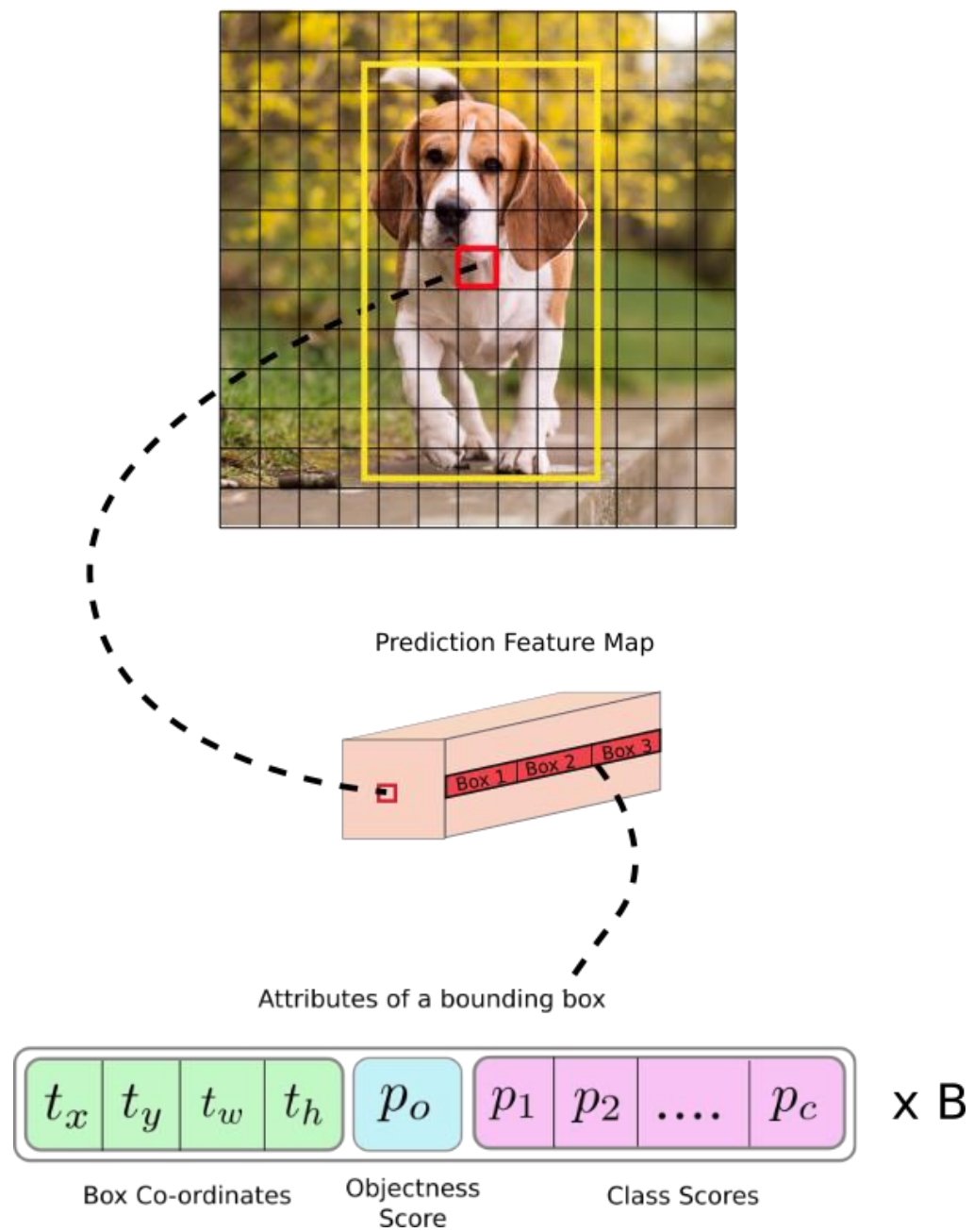
You Only Look Once: Unified, Real-Time Object Detection

- YOLO将特征图划分为 $S \times S$ 的格子(grid cells), 每个格子负责对落入其中的目标进行检测, 一次性预测所有格子所含目标的边界框、定位置信度、以及所有类别概率向量。



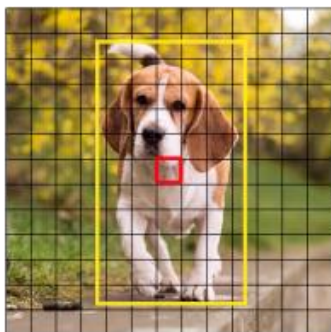
YOLO算法的基本思想



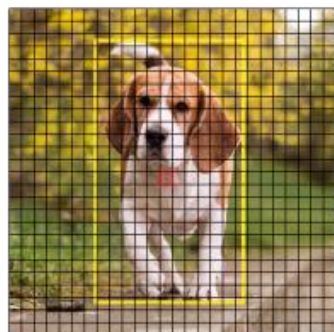


多尺度融合

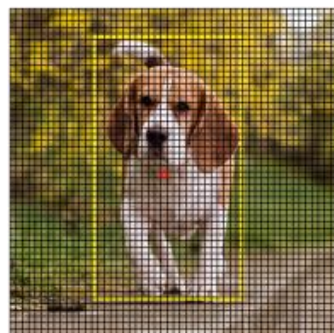
Prediction Feature Maps at different Scales



13 x 13

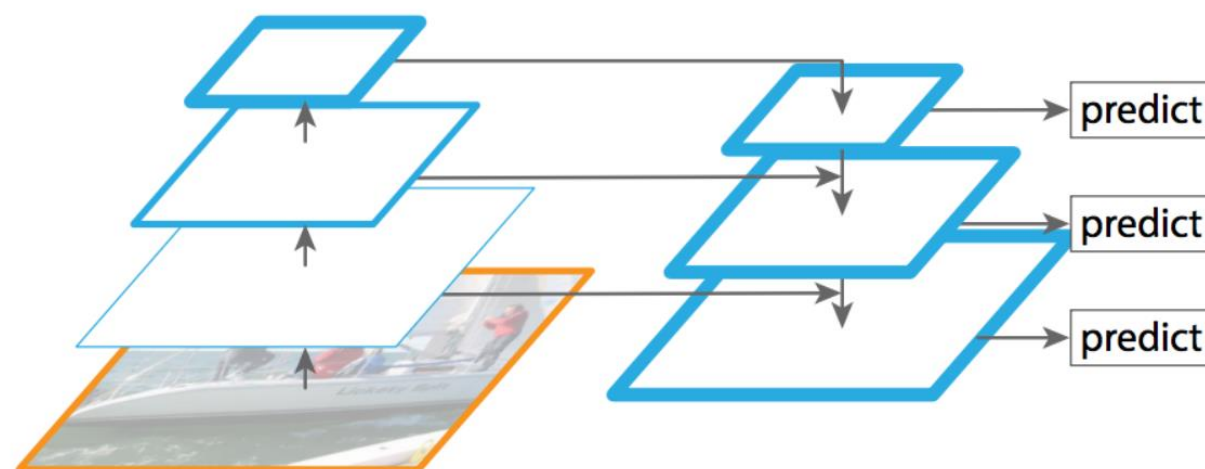


26 x 26

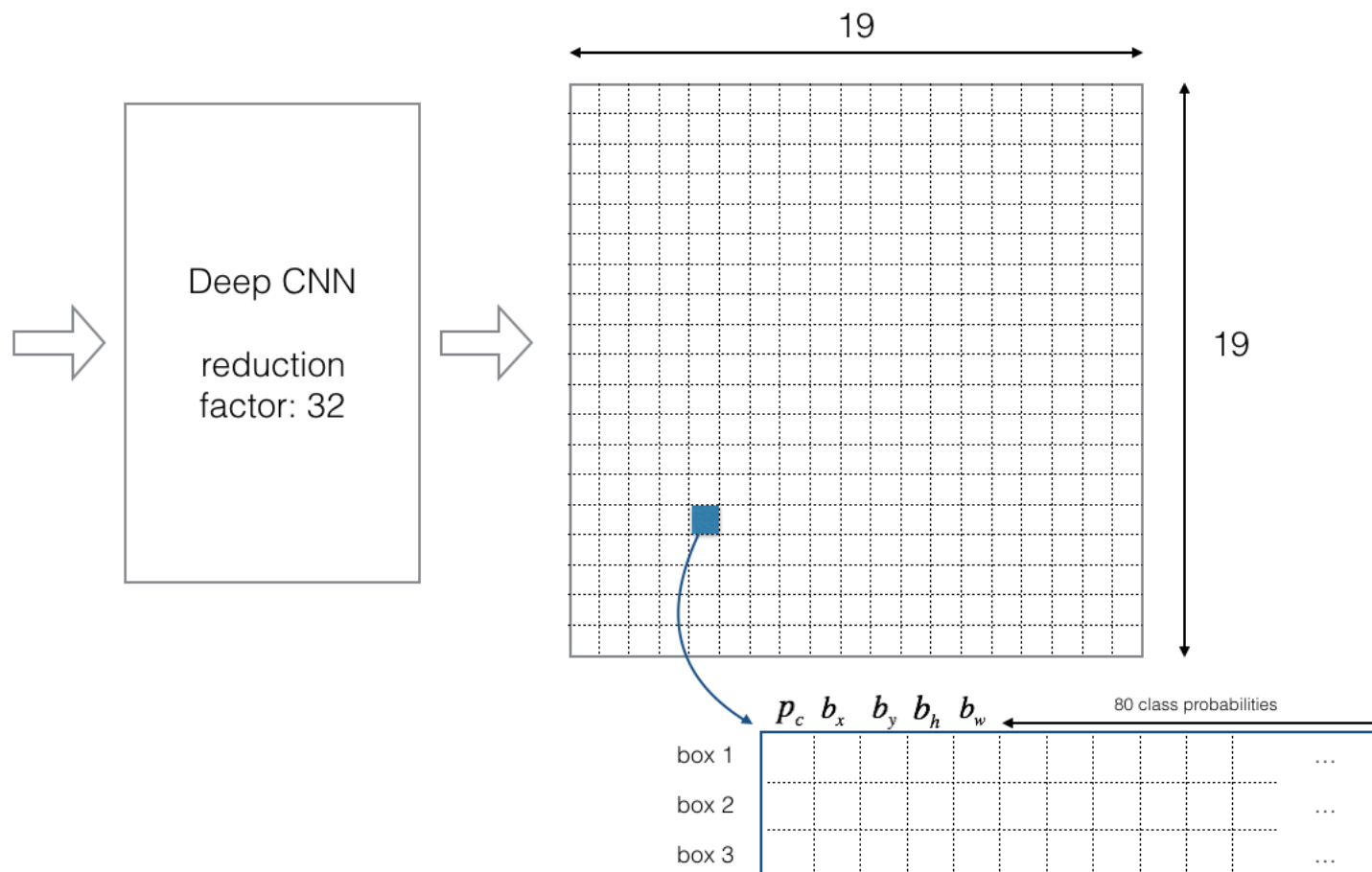


52 x 52

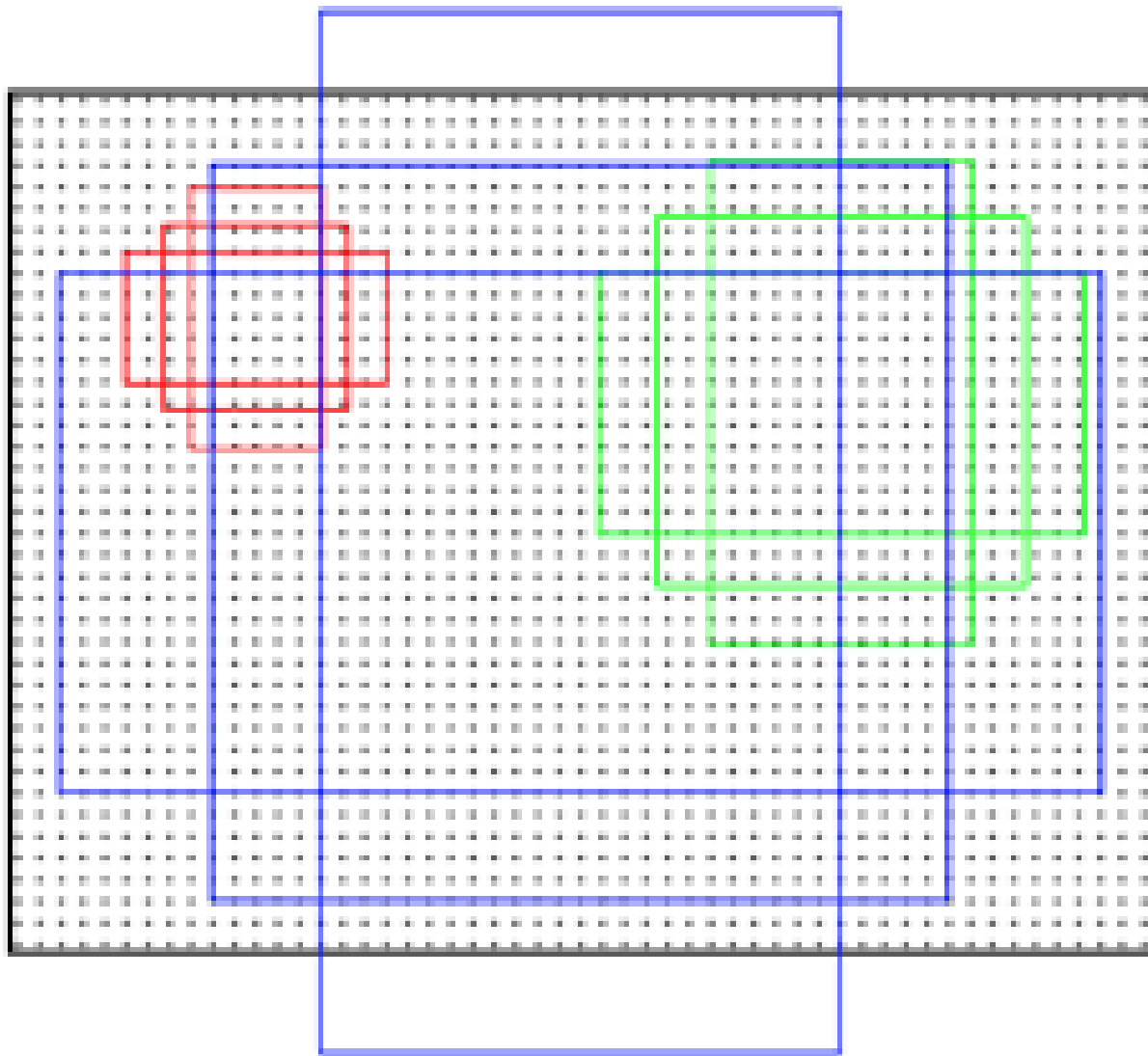
Feature Pyramid Network (FPN)



```
preprocessed image
(608, 608, 3)
```

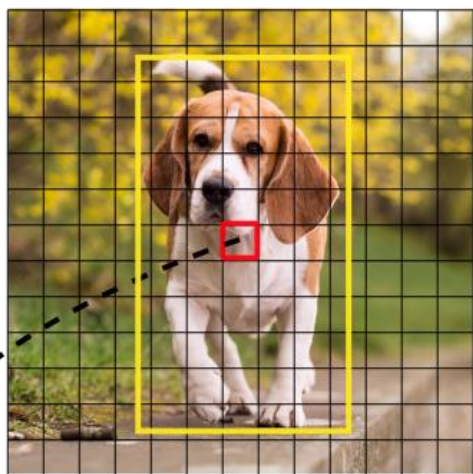


Anchor（锚框）机制

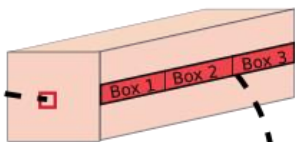


YOLO算法的基本思想

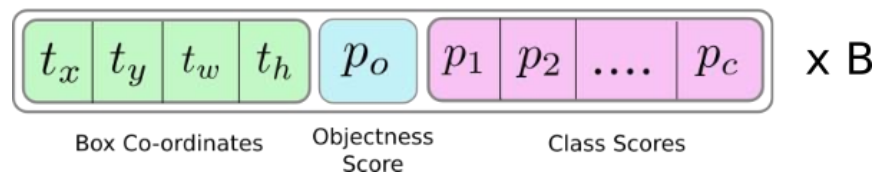
Image Grid. The Red Grid is responsible for detecting the dog



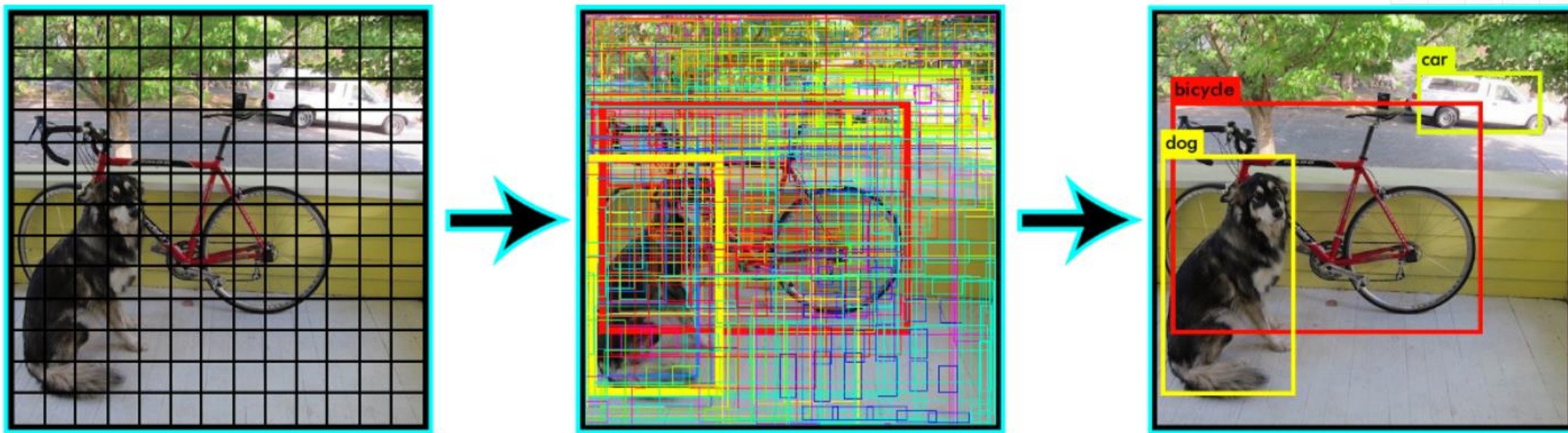
Prediction Feature Map



Attributes of a bounding box



- 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 13×13 （相当于 416×416 图片大小），然后将输入图像分成 13×13 个grid cells
 - YOLOv3/v4: 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。每个grid cell都会预测3个不同尺度的边界框。
 - YOLOv5: 不同于yolov3/v4，其GT可以跨层预测，即有些bbox在多个预测层都算正样本；匹配数范围可以是3-9个。
- 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 13×13 ，还有一个维度（深度）是 $B \times (5+C)$
 - 注：B表示每个grid cell预测的边界框的数量（YOLO v3/v4中是3个）；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）。



每个预测框的类别置信度得分(class confidence score) 计算如下:

$$\text{class confidence score} = \text{box confidence score} \times \text{conditional class probability}$$

它测量分类和定位（目标对象所在的位置）的置信度。

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

box confidence score $\equiv P_r(\text{object}) \cdot \text{IoU}$

conditional class probability $\equiv P_r(\text{class}_i | \text{object})$

class confidence score $\equiv P_r(\text{class}_i) \cdot \text{IoU}$

= box confidence score \times conditional class probability

where

$P_r(\text{object})$ is the probability the box contains an object.

IoU is the IoU (intersection over union) between the predicted box and the ground truth.

$P_r(\text{class}_i | \text{object})$ is the probability the object belongs to class_i given an object is presence.

$P_r(\text{class}_i)$ is the probability the object belongs to class_i

NMS (Non-Maximum Suppression)

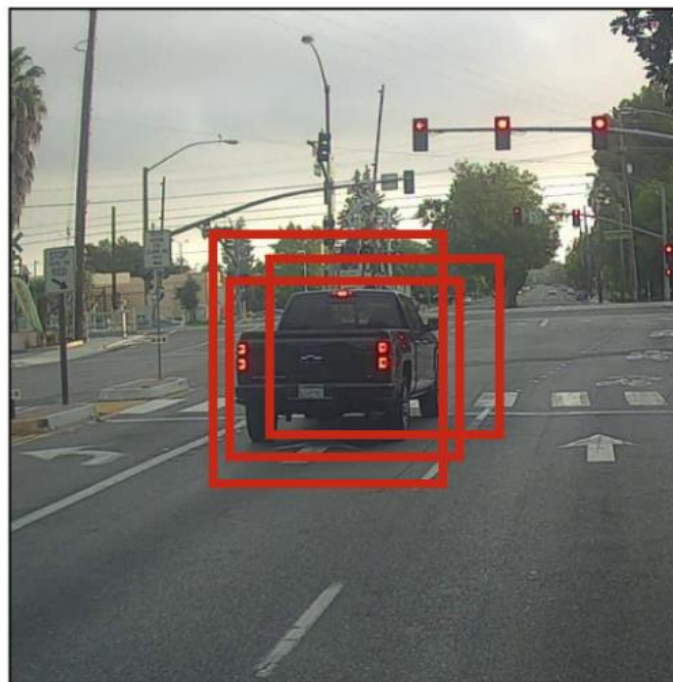
非极大抑制

测试时没有GT框，只能比较多个预测框，比较相互之间的IOU，做NMS



Multiple Grids may detect the same object
NMS is used to remove multiple detections

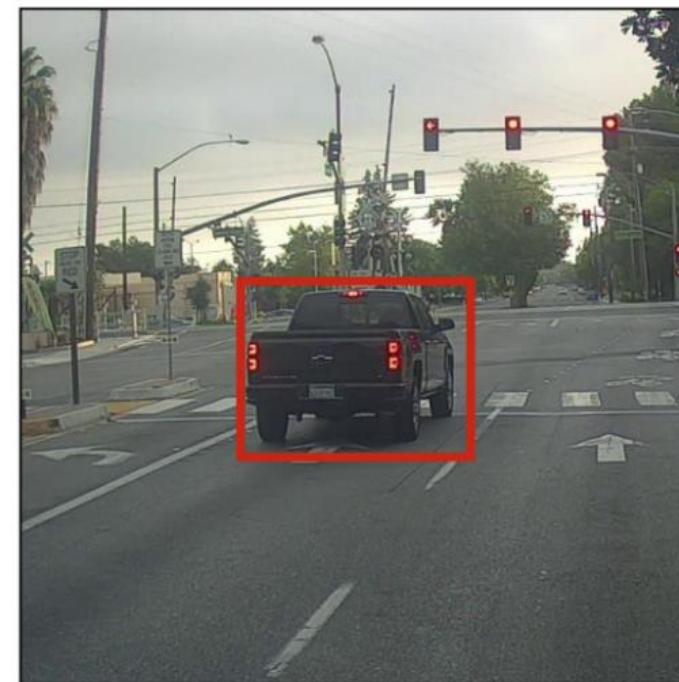
Before non-max suppression



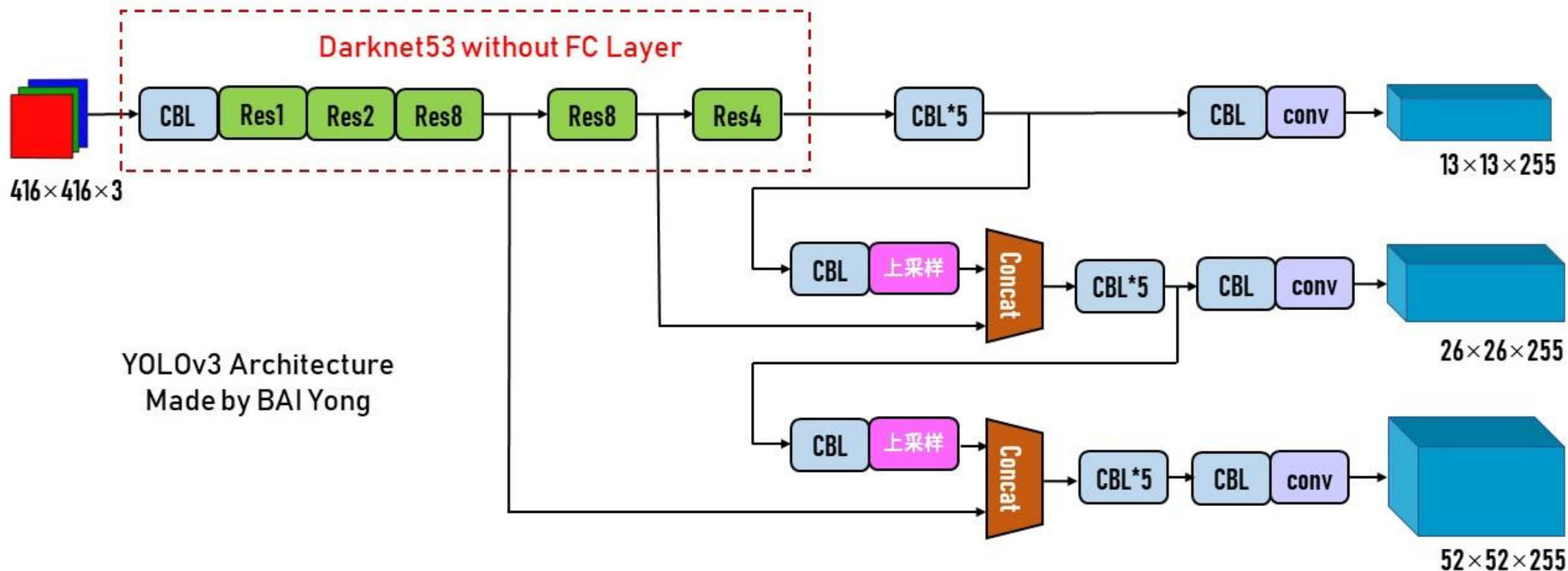
Non-Max
Suppression



After non-max suppression

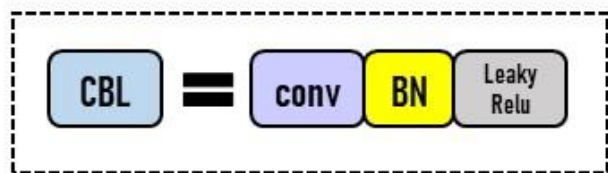


YOLOv3 网络架构

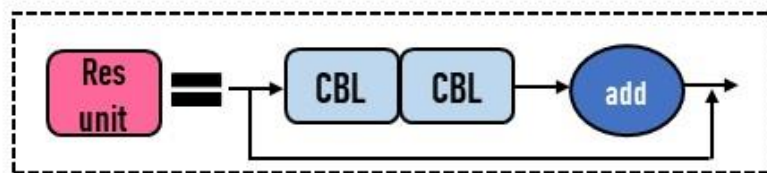


YOLOv3 Architecture
Made by BAI Yong

CBL (DarknetConv2D_BN_Leaky)



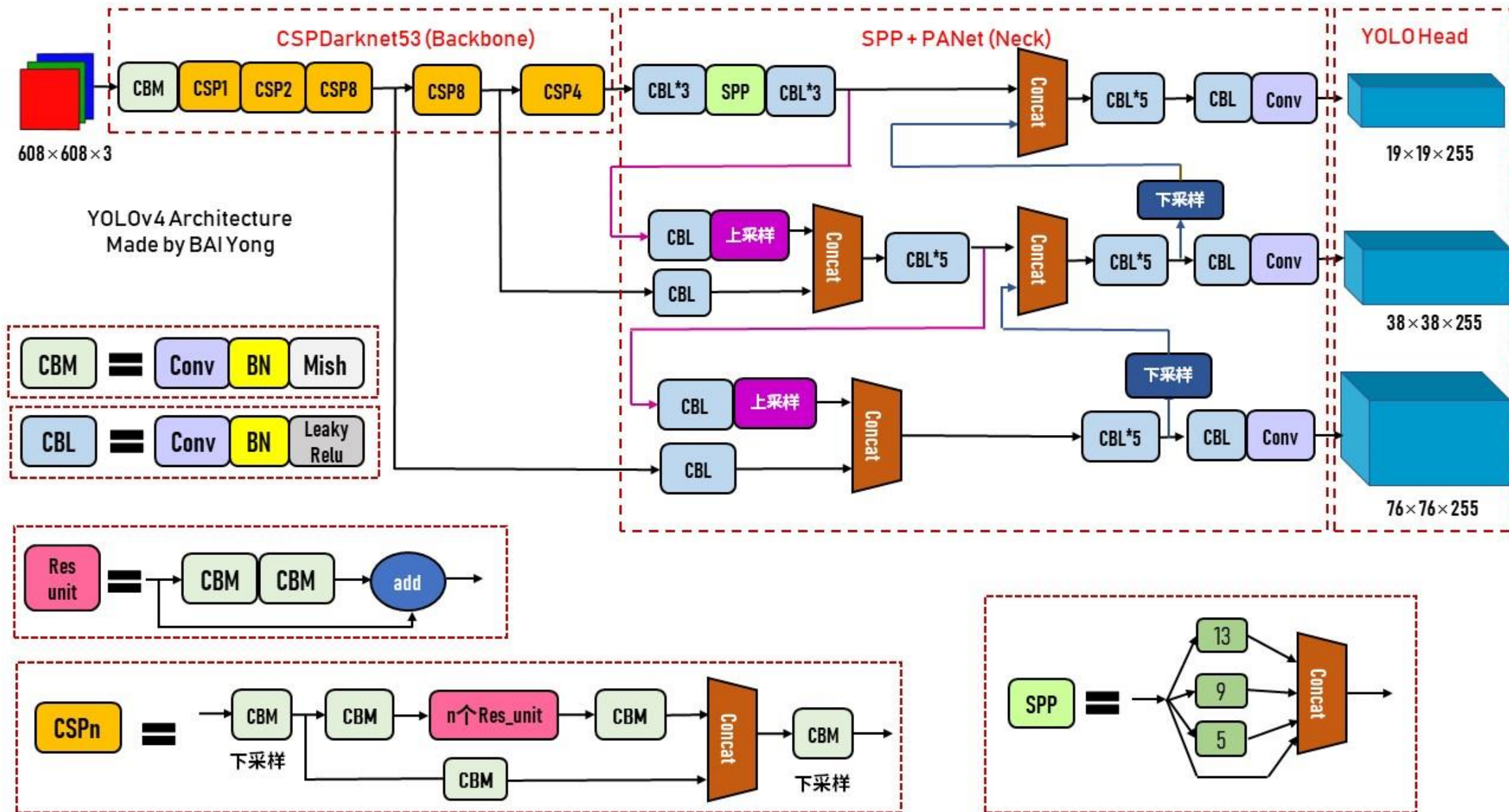
Res_unit



Resblock_body



YOLOv4 网络架构



损失函数(Loss function)

损失函数包括：

- **classification loss**, 分类损失
- **localization loss**, 定位损失（预测边界框与GT之间的误差）
- **confidence loss**, 置信度损失（框的目标性； objectness of the box）

总的损失函数：

classification loss + localization loss + confidence loss