

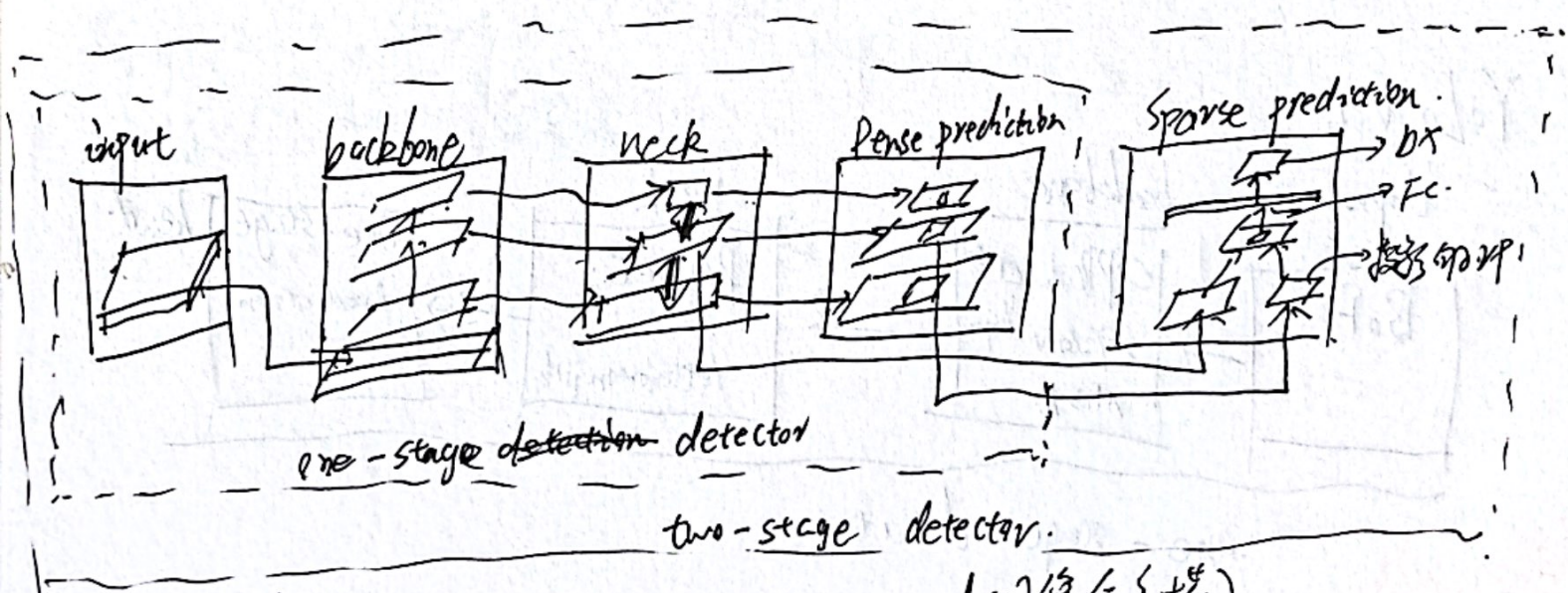
YOLO v4

v4 相较于 v3 的主要改进是采用了近年来 CNN 中领先的优化策略，追求低延迟 主要体现现在各种优化算法在 v3 结构上的尝试

1. 模型更高效且强大 兼顾了 FPS 与 precision 的平衡.
2. 验证一系列 state-of-the-art 目标检测用基训练方法的有效性.
3. 使用 DCBN, PAN, SAM 等方法提高 CPU 训练效率

两类训练策略:

1. Bag of freebies: 通过训练策略或者增加训练成本 (数据增强)
2. Bag of specials: 增加一点推理成本 (插件模式或后处理), 大幅提升检测精度.



Input: 输入 image, patches, input pyramid (图像金字塔).

Backbone: 各种变式 VGG16, ResNet-50, ResNext-101, Darknet-53, CSP DarkNet-53 等

neck:

additional blocks: SPP, ASPP, RFB, SAM.

Path-aggregation blocks: FPN, PAN, Mgs-FPN, Fully-Connected FPN.

BiFPN, ASiT, SFAM.

Head:

one-stage ~~go~~ Dense Predictions:

Anchor: RPN, SSP, YOLO, RetinaNet.

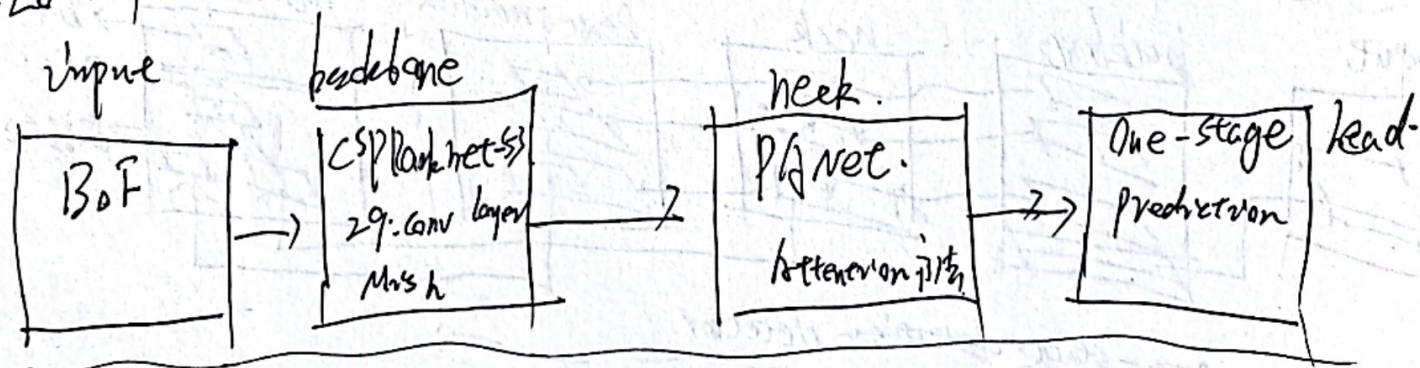
Free Anchor: CornerNet, CenterNet, MotusNet, FOS.

two-stage Sparse Predictions:

Anchor: Faster R-CNN, R-FCN, Mask R-CNN.

Free Anchor: RepPoints.

YOLO V4



one-stage detector.

V_4 相较于 V_3 节省了 head 部分. backbone 替换为 CSP Darknet 53. 采用
(空洞卷积于特征池化) 来扩大感受野 使用 PAFNet 作为 neck 部分

BoT (bag of freebies): 能够提高精度但不增加推断时间的技术

数据增强技术: 同位元内交换, cutout, grid mask 等.

网络正则化: Dropout, DropBlock 等.

损失函数优化.

Bas (bag of specials): 稍许增加推断代价但对精度有极大提升.

损失模型感受野: CFP, ACF, RFB 等.

3D Attention: SE, SAM 等.

特征提取层: SAM, ASFF, BiTPN.

改进激活函数: 与处理前

Yolo v5

V5 相较于 V4 的改进处.

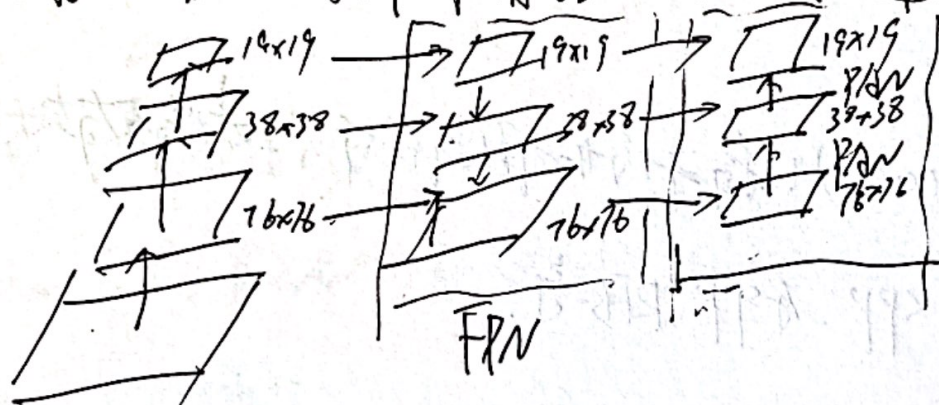
在 V4 中 CSP Darknet-53 只用于 backbone, V5 将两种 CSP 结构分别用于 backbone 与 neck.

CSP1-X

CSP2-X

V5 与 V4 的 Neck 结构对比. 在 Neck 的过程中采用不同的方法

neck:



V4 中采用普通的 conv 实现 neck 过程

V5 中采用 CSP2-X 实现 neck 中的过程

正则化 66x66 与 66x66
重叠面积

V5 中采用 GIOU

$$GIOU = \underbrace{IOU}_{\text{重叠}} - \frac{|A_c - U|}{|A_c|}$$

外边界包围面积