Weekly Report

2020.11.04

weihang

A SOLUTION TO PRODUCT DETECTION IN DENSELY PACKED SCENES

- 数据调整
  - 为了保证输入都是超高resolution level，把输入图片随机裁剪 **Random Seven Crop**
- detector 调整
  - 把**max positive sample number** of both RPN and R-CNN 的值都调大：**256 to 512**
  - **Cascade R-CNN**
- 其他
  - NMS hyper-paramter optimizing 用的grid search
  - 使用**RexNeXt**来代替ResNet50

A SOLUTION TO PRODUCT DETECTION IN DENSELY PACKED SCENES

Random Seven Crop We designed a strategy to relieve these two disadvantage. Clipping the bounding box may cause some fake box whose entity in box has been clipped out but background still remains. These fake box can lead to confusion to model in training. Hence we only remain a clipped box whose IoU to origin box higher than a threshold. Regular random crop sample the position of crop region from a uniform distribution. Random Seven Crop is designed to sample the region from only seven certain position: Four corners of image, center point and two end points of short axis.

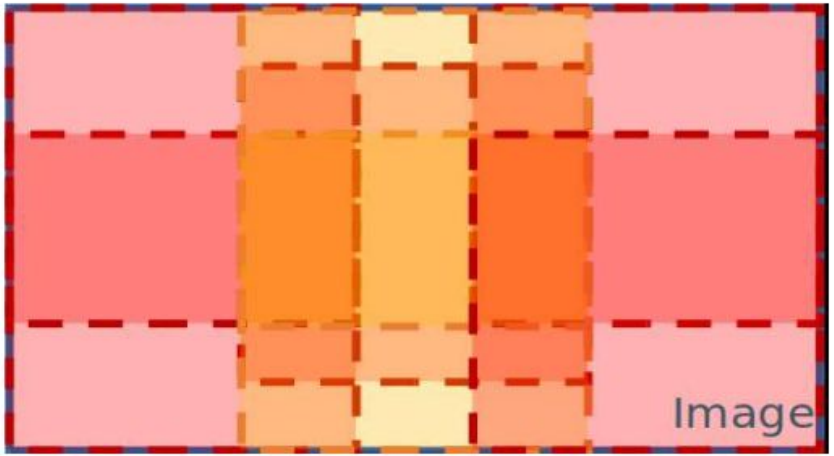
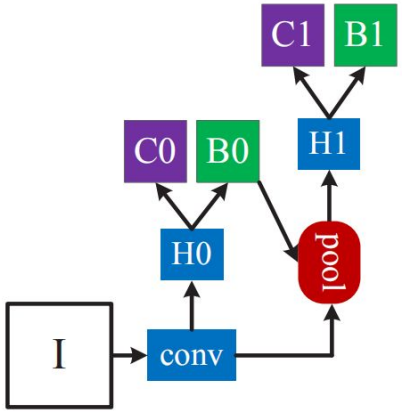


Figure 2: The seven sample areas of Random Seven Crop

Faster R-CNN



(a) Faster R-CNN

BBox
回归损失

$f(x, \mathbf{b})$

$$\mathcal{R}_{loc}[f] = \sum_{i=1}^{N} L_{loc}(f(x_i, \mathbf{b}_i), \mathbf{g}_i), \quad (1)$$

where L_{loc} was a L_2 loss function in R-CNN [12], but updated to a smoothed L_1 loss function in Fast-RCNN [11]. To encourage a regression invariant to scale and location, L_{loc} operates on the distance vector $\Delta = (\delta_x, \delta_y, \delta_w, \delta_h)$

$$\delta_x = (g_x - b_x)/b_w, \quad \delta_y = (g_y - b_y)/b_h$$
$$\delta_w = \log(g_w/b_w), \quad \delta_h = \log(g_h/b_h). \quad (2)$$

Class
分类损失

$h(x)$

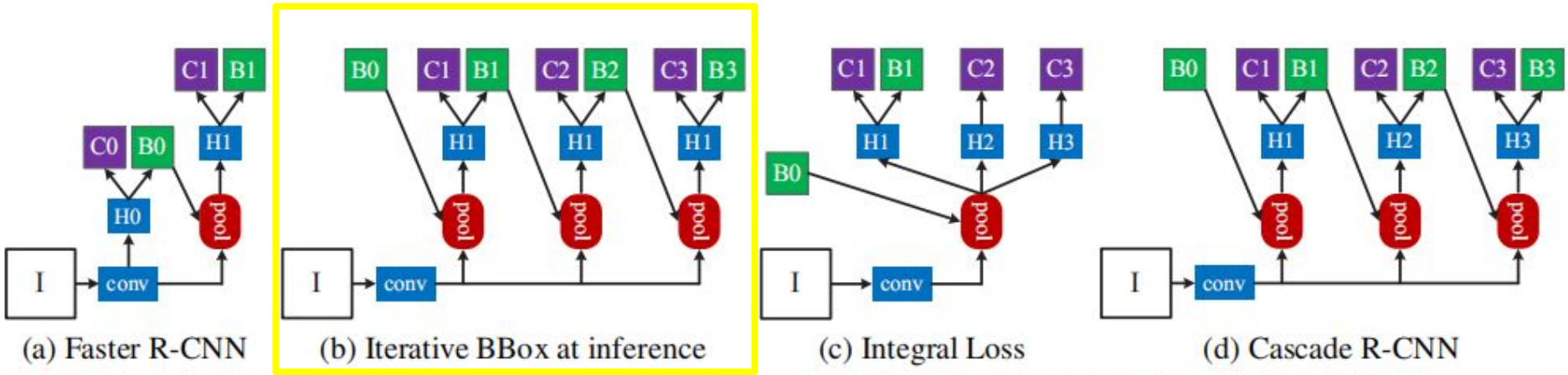
background and the remaining the objects to detect. $h(x)$ is a $M + 1$-dimensional estimate of the posterior distribution over classes, i.e. $h_k(x) = p(y = k|x)$, where y is the class label. Given a training set (x_i, y_i), it is learned by minimizing a classification risk

$$\mathcal{R}_{cls}[h] = \sum_{i=1}^{N} L_{cls}(h(x_i), y_i), \quad (4)$$

where L_{cls} is the classic cross-entropy loss.

Figure 3. The architectures of different frameworks. "I" is input image, "conv" backbone convolutions, "pool" region-wise feature extraction, "H" network head, "B" bounding box, and "C" classification. "B0" is proposals in all architectures.

Cacade R-CNN



(a) Faster R-CNN (b) Iterative BBox at inference (c) Integral Loss (d) Cascade R-CNN

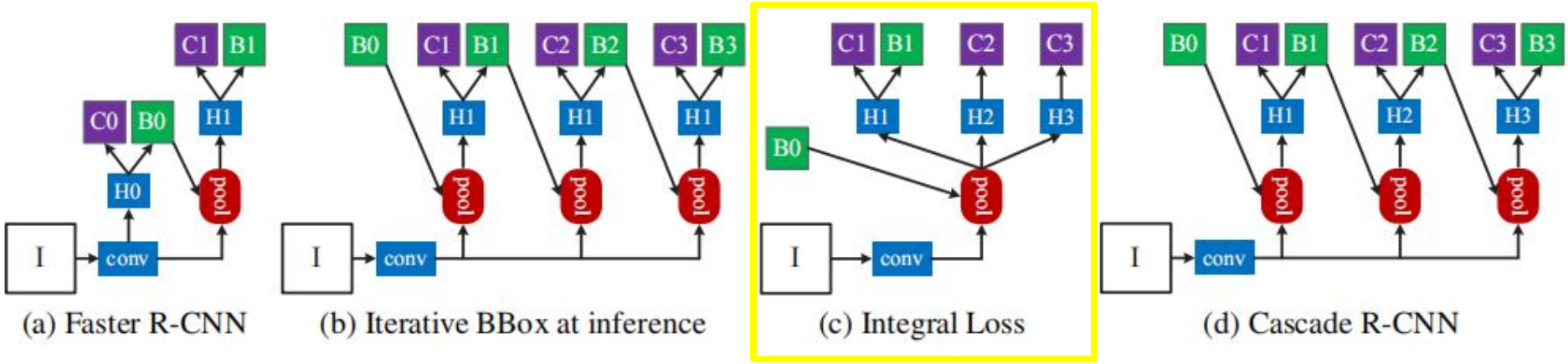
stead, f is applied iteratively, as a post-processing step

$$f'(x, \mathbf{b}) = f \circ f \circ \cdots \circ f(x, \mathbf{b}), \quad (3)$$

to refine a bounding box $\mathbf{b}$. This is called *iterative bounding box regression*, denoted as *iterative BBox*. It can be

用同一个f迭代多次

Cacade R-CNN



(a) Faster R-CNN (b) Iterative BBox at inference (c) Integral Loss (d) Cascade R-CNN

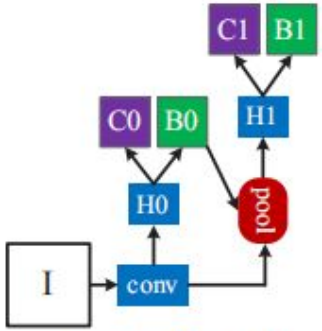
优化Detector，是在优化它对于TP和FP的辨别力，即IoU threshold设定的挣扎。
那么问题来了，这个IoU threshold怎么设置都不合理，怎么办呢 -- **用ensemble**

$$L_{cls}(h(x), y) = \sum_{u \in U} L_{cls}(h_u(x), y_u), \quad (6)$$

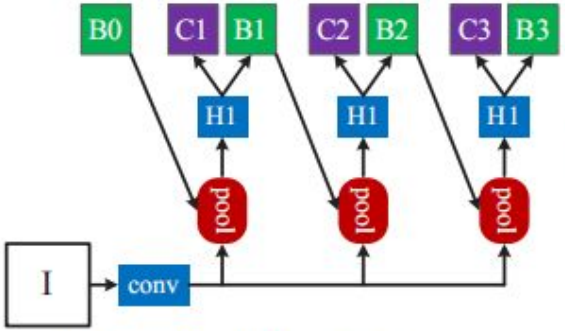
where U is a set of IoU thresholds. This is closely

只有一个f，有多个h

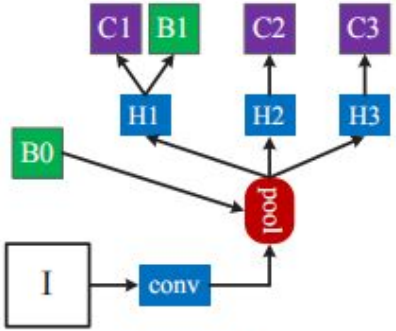
Cacade R-CNN



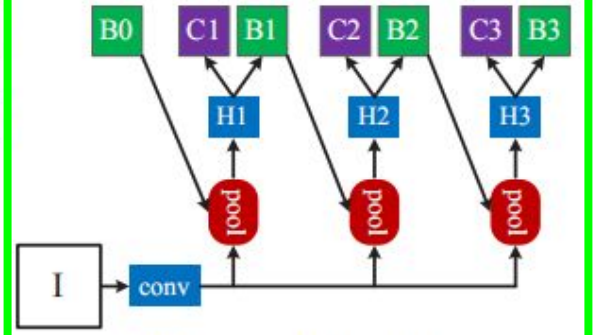
(a) Faster R-CNN



(b) Iterative BBox at inference



(c) Integral Loss



(d) Cascade R-CNN

cascade of *specialized* regressors

$$f(x, \mathbf{b}) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x, \mathbf{b}), \quad (7)$$

At each stage t, the R-CNN includes a classifier h_t and a regressor f_t optimized for IoU threshold u^t, where $u^t > u^{t-1}$. This is guaranteed by minimizing the loss

$$L(x^t, g) = L_{cls}(h_t(x^t), y^t) + \lambda[y^t \geq 1] L_{loc}(f_t(x^t, \mathbf{b}^t), \mathbf{g}), \quad (8)$$

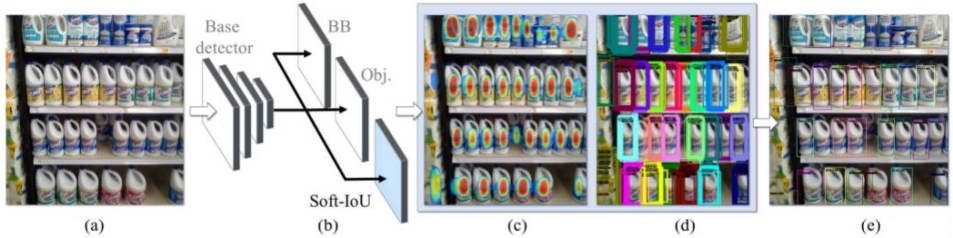
where $\mathbf{b}^t = f_{t-1}(x^{t-1}, \mathbf{b}^{t-1})$, g is the ground truth object for x^t, $\lambda = 1$ the trade-off coefficient, $[\cdot]$ the indicator function, and y^t is the label of x^t given u^t by (5). Unlike the

It differs from the *iterative BBox* architecture of Figure 3 (b) in several ways. First, while *iterative BBox* is a post-processing procedure used to improve bounding boxes, cascaded regression is a *resampling* procedure that changes the distribution of hypotheses to be processed by the different stages. Second, because it is used at both training and inference, there is no discrepancy between training and inference distributions. Third, the multiple specialized regressors $\{f_T, f_{T-1}, \cdots, f_1\}$ are optimized for the *resampled distributions* of the different stages. This opposes to the single f of (3), which is only optimal for the initial distribution. These differences enable more precise localization than *iterative BBox*, with no further human engineering.

Cacade R-CNN的思考

- 层层递进，数据（IoU）在变
- 层层递进，每一级的网络也在变（结构一样）
- 相当于对不同的样本训练了不同的分类器
- 这非常像boost

Precise Detection in Densely Packed Scenes

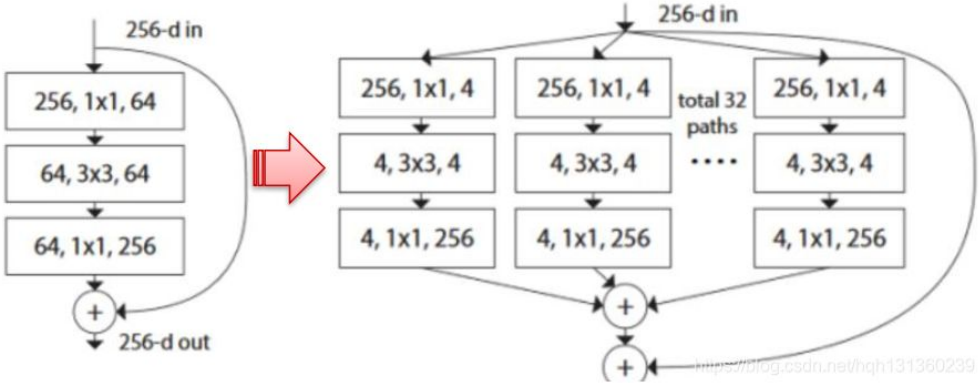


$$\mathcal{L}_{\text{sIoU}} = \qquad (2)$$
$$-\frac{1}{n}\sum_{i=1}^{n}[IoU_i \log(c_i^{iou}) + (1-IoU_i)\log(1-c_i^{iou})],$$

这些分布如果用聚类处理下呢？？

Futher Reading

ResNetXt



- 目标框的回归线grid
- 底部加loss
- 下载SKU数据集
- point net 中心点回归分类
- 细分类
- CRF
- NMS的改进