Headline

大家好：

这是**2018年度第2篇Arxiv Weekly**。

本文是检测方向的文章。

Highlight

> 利用**IoU-Net**提升**NMS**选框的性能，并利用新的方法微调选出的**bbox**最终提高检测性能。

Information

Title

Acquisition of Localization Confidence for Accurate Object Detection

Link

https://arxiv.org/pdf/1807.11590.pdf

Source

- 北京大学（Peking University）
- 清华大学（Tsinghua University）
- 旷视科技 （Face++）
- 头条人工智能实验室（Toutiao AI Lab）

Introduction

现代CNN目标检测器中，一般都存在 Non-maximum suppression (NMS) 选框和bbox回归的过程。然而这里面的逻辑是有一定的问题的。 NMS操作的置信度本质上来源于classification的label，而不是localization，这不可避免地导致了一些localization性能的降退，即使后续有进一步的回归。

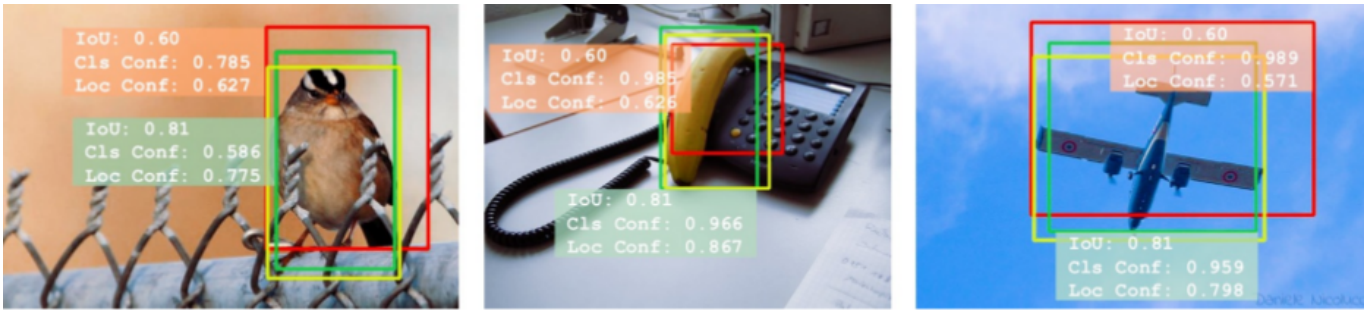
本文提出了IoU-Net结构，能够预测每个候选框和对应的ground truth框之间的IoU。由于在选框的时候真正考虑了localization信息，IoU-Net能够优化NMS过程，给出更精准的预选框。另外，本文也提出了bbox refinement算法，来进一步提高最终框的精度。

在MSCOCO上的实验表明本文提出的方法能在经典detection pipeline上取得涨点，并且具有良好的兼容性和可迁移性。

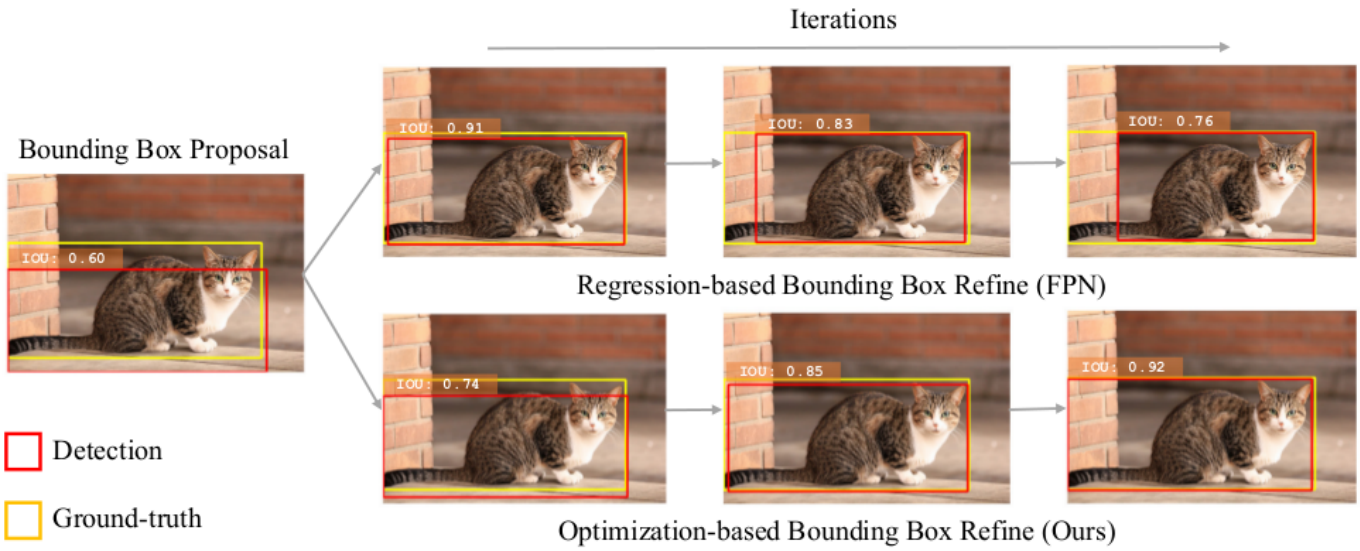
Keys

1.问题描述

- misalignment between classification and localization accuracy. [也即NMS会舍弃更好的框，挑选烂框来回归]



- the absence of localization confidence makes bbox regression less interpretable. [也即随着迭代的进行，bbox的回归结果反而恶化]



2.解决方案part1：IoU-guided NMS

Algorithm 1 IoU-guided NMS. Classification confidence and localization confidence are disentangled in the algorithm. We use the localization confidence (the predicted IoU) to rank all detected bounding boxes, and update the classification confidence based on a clustering-like rule.

Input: $\mathcal{B} = \{b_1, ..., b_n\}$, $\mathcal{S}$, $\mathcal{I}$, Ω_{nms}
$\mathcal{B}$ is a set of detected bounding boxes.
$\mathcal{S}$ and $\mathcal{I}$ are functions (neural networks) mapping bounding boxes to their classification confidence and IoU estimation (localization confidence) respectively.
Ω_{nms} is the NMS threshold.
Output: $\mathcal{D}$, the set of detected bounding boxes with classification scores.

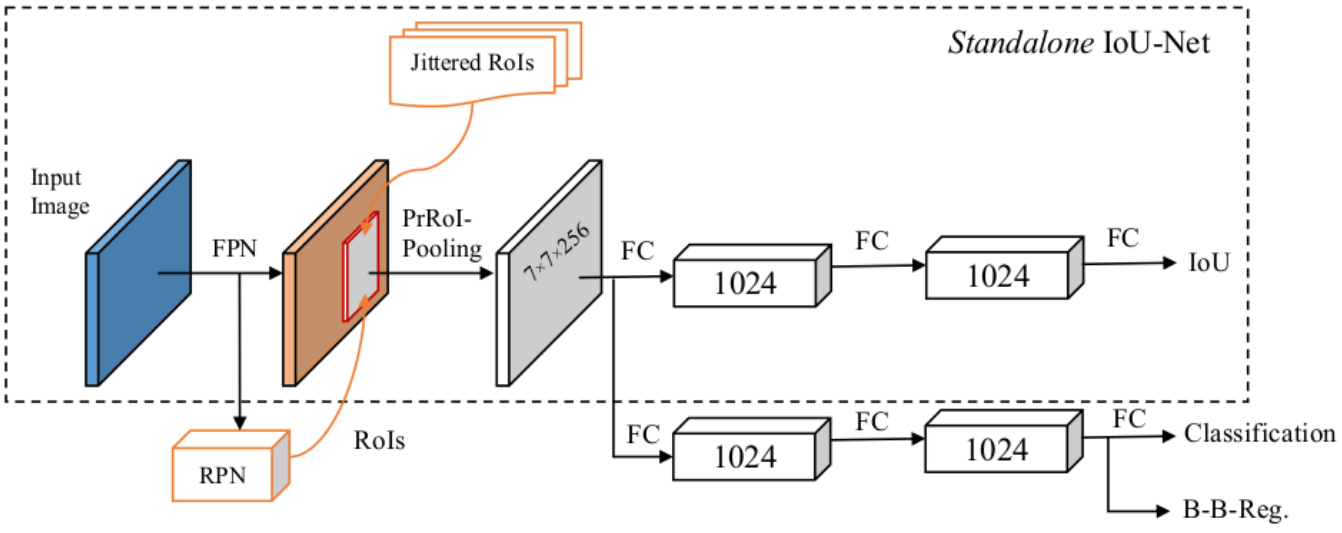
$\mathcal{D} \leftarrow \varnothing$
while $\mathcal{B} \neq \varnothing$ **do**
  $b_m \leftarrow \arg\max \mathcal{I}(b_j)$
  $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_m\}$
  $s \leftarrow \mathcal{S}(b_m)$
  for $b_j \in \mathcal{B}$ **do**
    if $\text{IoU}(\text{b}_\text{m}, \text{b}_\text{j}) > \Omega_{\text{nms}}$ **then**
      $s \leftarrow max(s, \mathcal{S}(b_j))$
      $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_j\}$
    end if
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\langle b_m, s \rangle\}$
end while
return $\mathcal{D}$

此为文中提出的新NMS算法，其核心为利用bbox对应区域和真值的IoU作为挑选maximum bbox的依据，而不是像传统的NMS一样直接用classification score来作为依据，这就弥合了misalignment between classification and localization accuracy。

另外值得一提的是，IoU-guided NMS算法中，在抑制那些和最大框交集超过阈值的框的同时，会参考它们的分类信息。如果它们的分类置信度高于我们选出的最大框，则会更新最大框的分类。[这个操作中，分离classification和localization的意味也很重，甚至有点localization指导优化classification的意思在里面。不过个人感觉这种情况并不常见]

剩下的一个关键问题是，给出一个bbox，我们怎么知道它和真值的bbox的IoU是多少……

文章中采用了如下的IoU-Net网络来解决这个问题。



网络的训练数据是通过ground truth + augmentation & randomization生成的。

3.解决方案part2：optimization based bounding box refinement

Algorithm 2 Optimization-based bounding box refinement

Input: $\mathcal{B} = \{b_1, ..., b_n\}$, $\mathcal{F}$, T, λ, Ω_1, Ω_2
$\mathcal{B}$ is a set of detected bounding boxes, in the form of (x_0, y_0, x_1, y_1).
$\mathcal{F}$ is the feature map of the input image.
T is number of steps. λ is the step size, and Ω_1 is an early-stop threshold and $\Omega_2 < 0$ is an localization degeneration tolerance.
Function PrPool extracts the feature representation for a given bounding box and function IoU denotes the estimation of IoU by the IoU-Net.
Output: The set of final detection bounding boxes.

$\mathcal{A} \leftarrow \varnothing$
for $i = 1$ *to* T **do**
  for $b_j \in \mathcal{B}$ and $b_j \notin \mathcal{A}$ **do**
    $\boldsymbol{grad} \leftarrow \nabla_{b_j}\text{IoU}(\text{PrPool}(\mathcal{F}, b_j))$
    $PrevScore \leftarrow \text{IoU}(\text{PrPool}(\mathcal{F}, b_j))$
    $b_j \leftarrow b_j + \lambda * scale(\boldsymbol{grad}, b_j)$
    $NewScore \leftarrow \text{IoU}(\text{PrPool}(\mathcal{F}, b_j))$
    if $|PrevScore - NewScore| < \Omega_1$ **or** $NewScore - PrevScore < \Omega_2$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{b_j\}$
    end if
  end for
end for
return $\mathcal{B}$

核心的思路在于利用IoU Net对bbox的IoU预测能力，指导后续的bbox微调。微调的时候用最简单的梯度上升优化方法，并设定两个阈值，一个用来判定收敛；一个用来判定是否已经进入localization degeneration的阶段并予以遏制。

p.s. 我们可以看到，不同的refinement方法，都是在寻找一种方式解决如下的最优化问题：

$$c^* = \arg\min_c crit(\mathbf{transform}(box_{\text{det}}, c), box_{\text{gt}}) , \qquad (1)$$

其中det是检测网络输出的bbox；gt是真值bbox；c是网络微调器transfomer的参数；crit是loss函数形式[一般选取smooth-L1 distance]

4.关于precise ROI pooling

为了能够实现3中的算法，我们需要给出一个可导的RoI Pooling算法。RoI Align算法已经比较好地解决了RoI中misalignment的问题，因此其基本想法是应该沿用的。所需要解决的是导数求解的问题。

因此本文提出了精准RoI Pooling (PrRoI Pooling)算法，其实质就是把feature map上的点利用双线性插值(bilinear interpolation)算法转化为连续的函数，如下公式所示：

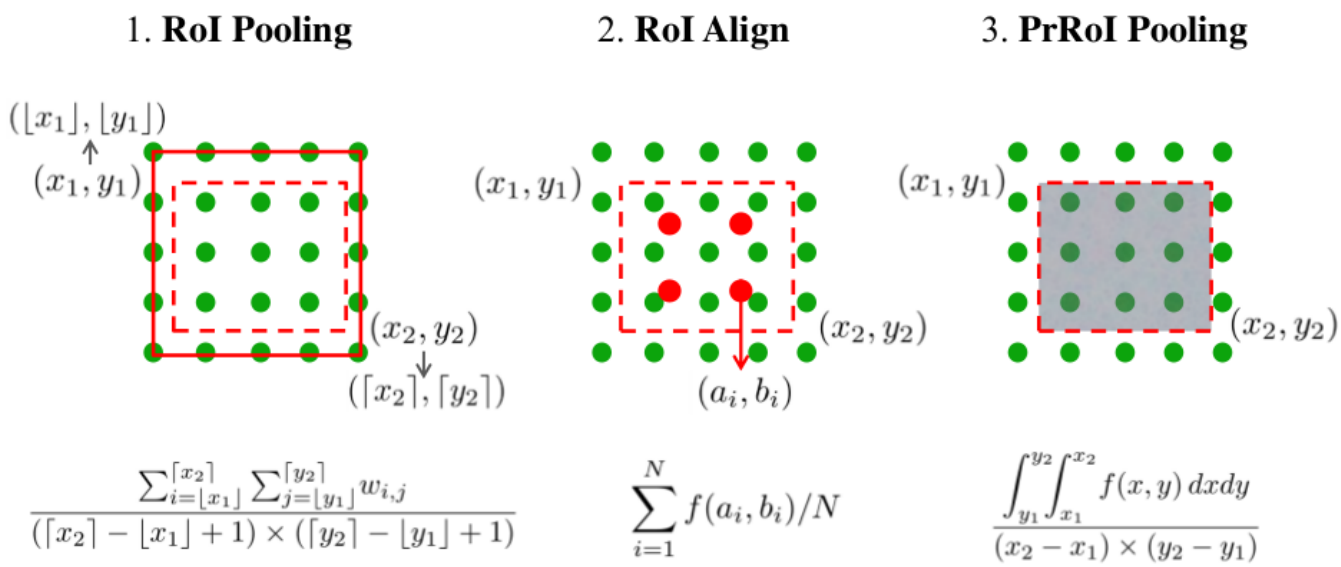
$$f(x,y) = \sum_{i,j} IC(x,y,i,j) \times w_{i,j}, \tag{2}$$

where $IC(x,y,i,j) = max(0, 1-|x-i|) \times max(0, 1-|y-j|)$ is the interpolation coefficient. Then denote a bin of a RoI as $bin = \{(x_1,y_1),(x_2,y_2)\}$, where (x_1,y_1)

有了连续的feature map值，就能够把bbox以浮点数的形式直接套在feature map对应的位置上，并且进行pooling操作了。[不过由于连续性的操作，我理解这里的PrRoI Pooling更加像一种average pooling的变体]

$$\mathrm{PrPool}(bin, \mathcal{F}) = \frac{\int_{y1}^{y2}\int_{x1}^{x2} f(x,y)\,dxdy}{(x_2-x_1)\times(y_2-y_1)}. \tag{3}$$

PrRoI Pooling和两种经典Pooling算法的对比示意图如下：



Results

所有实验均在MSCOCO上进行。

传统NMS、Soft-NMS、IoU-NMS的对比里面，高阈值AP下IoU-NMS表现最为突出，这是因为IoU-NMS能够最大程度地优化具体的边界位置。

Method	+Soft-NMS	+IoU-NMS	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
FPN			36.4	**58.0**	**53.1**	44.9	31.2	9.8
	✓		36.8	57.5	**53.1**	**45.7**	32.3	10.3
		✓	**37.3**	56.0	52.2	45.6	**33.9**	**13.3**
Cascade R-CNN			40.6	**59.3**	55.2	49.1	38.7	16.7
	✓		**40.9**	58.2	**54.7**	**49.4**	**39.9**	17.8
		✓	40.7	58.0	**54.7**	49.2	38.8	**18.9**
Mask-RCNN			37.5	**58.6**	**53.9**	46.3	33.2	10.9
	✓		37.9	58.2	**53.9**	**47.1**	34.4	11.5
		✓	**38.1**	56.4	52.7	46.7	**35.1**	**14.6**

将refinement操作附加在主流的pipeline上，都取得了不错的涨点。同样地，高阈值下涨点突出很多。

Method	+Refinement	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
FPN		36.4	**58.0**	**53.1**	44.9	31.2	9.8
	✓	**38.0**	57.7	**53.1**	**46.1**	**34.3**	**14.6**
Cascade R-CNN		40.6	**59.3**	55.2	49.1	38.7	16.7
	✓	**41.4**	**59.3**	**55.3**	**49.6**	**39.4**	**19.5**
Mask-RCNN		37.5	**58.6**	**53.9**	46.3	33.2	10.9
	✓	**39.2**	57.9	53.6	**47.4**	**36.5**	**16.4**

以上方法联合使用效果更佳。

Backbone	Method	+IoU-NMS	+Refine	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
ResNet-50	FPN			36.4	58.0	53.1	44.9	31.2	9.8
	IoU-Net			37.0	**58.3**	**53.8**	45.7	31.9	10.7
		✓		37.6	56.2	52.4	46.0	34.1	14.0
		✓	✓	**38.1**	56.3	52.4	**46.3**	**35.1**	**15.5**
ResNet-101	FPN			38.5	**60.3**	**55.5**	47.6	33.8	11.3
	IoU-Net			38.9	60.2	**55.5**	47.8	34.6	12.0
		✓		40.0	59.0	55.1	48.6	37.0	15.5
		✓	✓	**40.6**	59.0	55.2	**49.0**	**38.0**	**17.1**

Insights

针对检测中通用的NMS模块提出了新的改进，并且提出了在最后对bbox进行refinement操作的想法。比较好地缓解了分类和定位标准误用以及随着训练迭代bbox反而退化的问题。

另外，提出了对ROI Pooling求导数的一种思路：PrRoI Pooling。