

SSD 算法以及 Faster-RCNN 实现通过 X 光 图片检测充电宝

一. 问题描述

需要训练目标检测模型，使用这个目标检测模型检测出测试集中每张图片中的危险品（例如输入模型一张测试图片，最终输出这张图片中所有危险品的类别以及位置坐标）。在最后测试结果时，会使用测试集来进行测试，测试集和训练集的文件结构是相同的，但测试集没有给出，所以请从训练集中划分一个验证集出来。在验证集上验证自己模型的效果。

二. 任务一（遮挡问题）

数据集中的安检图片都存在不同等级的遮挡问题，遮挡会严重影响检测器识别危险品的准确率。如何解决严重遮挡条件下模型检测危险品问题是一个热点。

安检机返回的 x 光图像为 RGB 彩色图像。训练集中的危险品包括带电芯充电宝和不带电芯充电宝两个类别。训练集中共有 6000 张图片（带电芯充电宝和不带电芯充电宝各 3000 张，且测试集根据遮挡等级分为了 1,2,3 种不同的遮挡等级，不同遮挡等级的示意图见图 1）。每张图片都拥有一个危险品所在的位置标注文件，标注文件里的每行表示（危险品的名称，危险品位置的左上坐标，危险品位置的右下坐标）。

最后将修改测试文件中的数据集路径，将其改为测试集所在的路径。之后分别得出 3 个遮挡等级各自的 map。根据 3 个 map 与对应遮挡等级的权重（等级 1 权重 0.2，等级 2 权重 0.3，等级 3 权重 0.5）相乘后得到的分数相加得出最终的分数。

三. 问题分析

在本题中，由于数据集中分类数量有限，只有两处，且经过数据集的筛查发现，所谓的遮挡问题是发生在被分类物品与其他事物中，且遮挡物无需标定，即遮挡物消弱的是特征提取的准确性和丰富性，并不会由于遮挡而导致该物品被错分为别的类（例如：A 遮挡 B，由于 B 是部分，所以 B 被错认为 A。本数据集中的情况为：A 遮挡 B，B 只有部分特征，或者由于 A 覆盖 B 而导致特征存在噪声，从而影响特征提取和分类的准确性）。

从以上角度招手，更应该研究的是如何减弱噪声或者增强特征提取的准确性，本文采用了两种不同目标识别的方法，虽然没有做到进一步的优化，但通过尝试模型选择和调制参数学习了深度学习进行目标识别的整个流程，比较了两种目前主流的目标识别的方法，且考虑到识别目标所占大小偏小的特征，SSD 和 Faster-RCNN 分别采用两种策略对小目标进行识别，都有值得学习和借鉴的地方，也是本文想要说明的主要目的。

四. 作业流程

分别使用了 SSD 与 Faster-RCNN 两种算法进行目标检测，试图分析比较两种算法的异

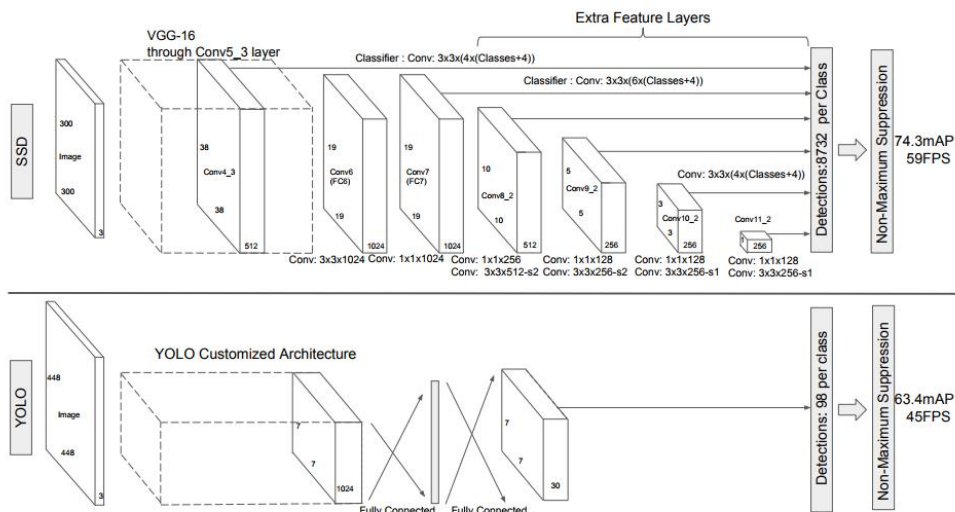
同以及在本数据集上的表现。在进行训练前，将数据集按照 VOC2007 的模式进行改变，之后进行训练，对比结果，改变参数优化模型。在训练 Faster-RCNN 时，将 Soft NMS 算法融合进去。

五. SSD 模型介绍

本项目主要采用的一种解决目标检测问题的方法为 SSD 模型，来源于 Wei Liu 的论文《SSD: Single Shot MultiBox Detector》，主要的思想和特点在于：

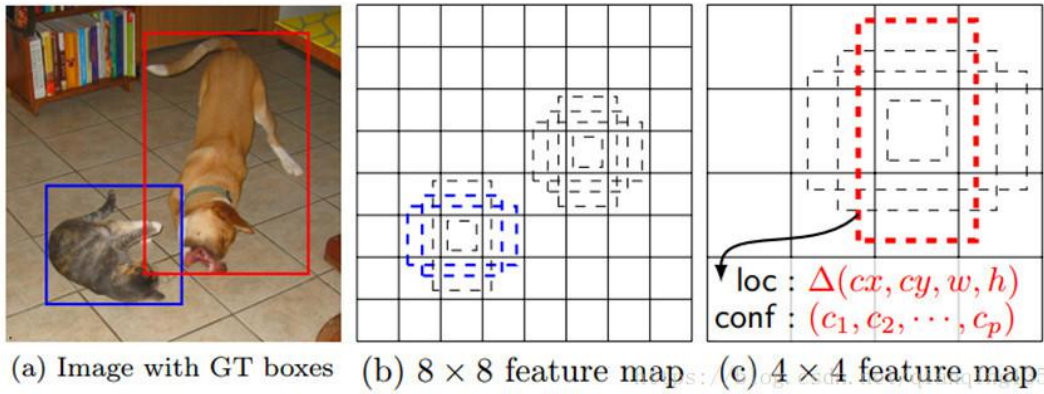
- 目标检测的定义理解为物体多个边界框如何进行有效回归的问题，在处理过程中会对网络框进行置信度得分设置，较高的置信度意味着该显示框匹配对象的可能性越高。
- 在模型优化和回归过程中，通过反向传播优化预测框和 Ground Truth 的相似度，从位置和分类两个角度进行分析和考虑。
- 增加边界框的可拓展型，设置多重比例的预选框，但在物体类别确定时，Bounding Box 的数量不会随之上升，易拓展于较大的数据集中。

1. 网络结构介绍



本项目在将 VGG16 作为 Base NetWork 的基础上，使用 6 个不同特征图检测不同尺度的目标，低层预测小目标，高层预测大目标。

2. 多种宽高比预测框



SSD 模型采用特征金字塔，从不同尺度的特征图下面来预测目标分类与位置。

在金字塔结构中每一部分都有 3×3 的卷积来进行预测，在某个位置上得到一个预测值，这个预测值可能是一个分类的得分，也可能是现对于默认框的位置偏差。从 SSD 网络结构图中可以看出来 conv6-2, conv7-2, conv8-2, conv9-2, fc7, conv4-2。

尺度线框的设定方式为：

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m - 1}(K - 1)$$

$$a_r \in 1, 2, 3, \frac{1}{2}, \frac{1}{3}$$

$$\omega_k^a = \sqrt{a_r} h_k^a = \frac{S_k}{\sqrt{a_r}}$$

3. 损失函数

损失函数定义为位置误差 (location loss, loc) 与置信度误差 (confidence loss, conf) 的加权和：

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

对于位置误差，其采用 Smooth L1 loss, 对于置信度误差，其采用 softmax loss。

4. 预测过程

预测过程中对于每个预测框，首先根据类别置信度确定其类别（置信度最大者）与置信度值，并过滤掉属于背景的预测框。

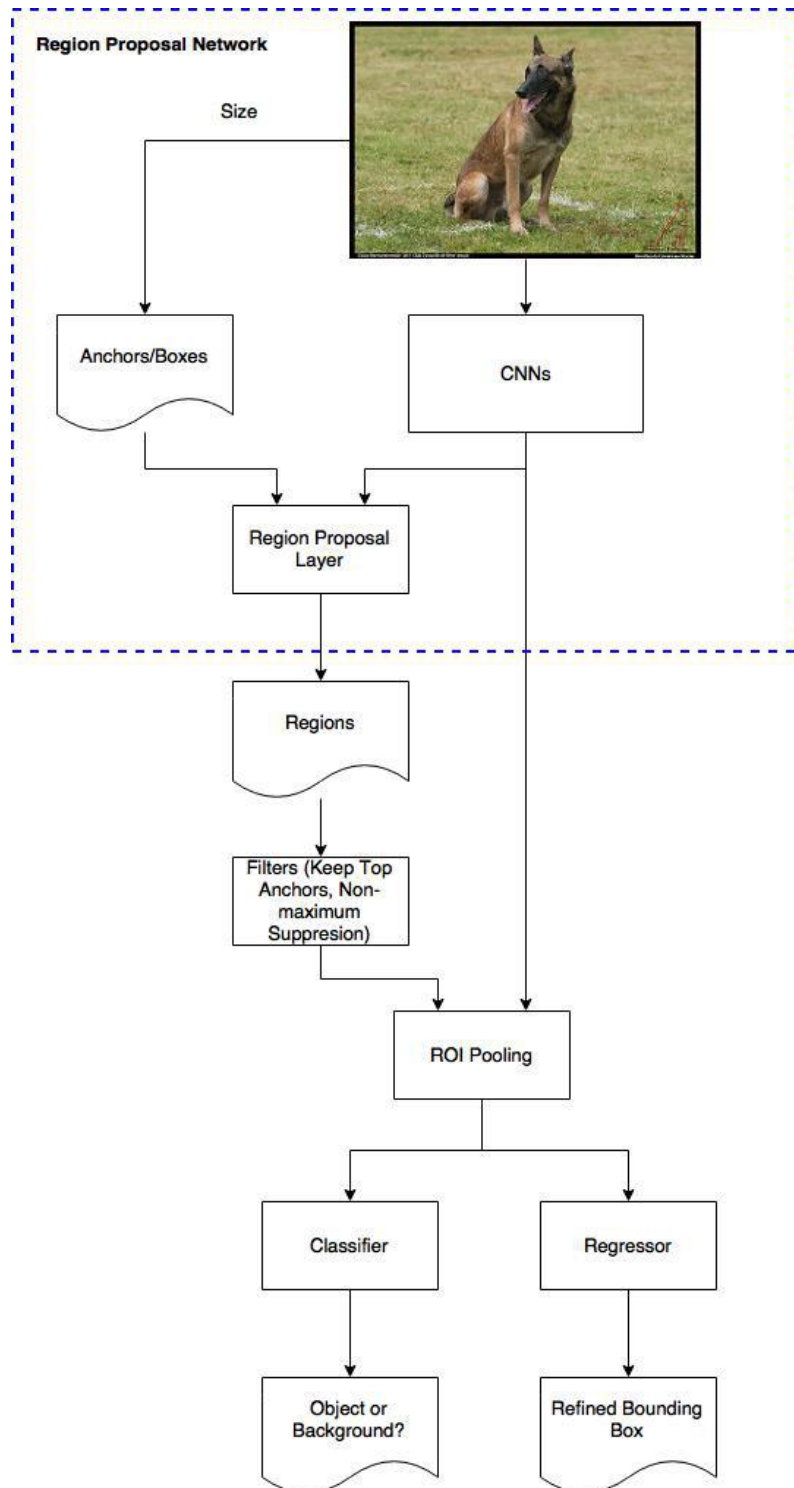
然后根据置信度阈值（如 0.5）过滤掉阈值较低的预测框。对于留下的预测框进行解码，根据先验框得到其真实的位置参数（解码后一般还需要做 clip，防止预测框位置超出图片）。解码之后，一般需要根据置信度进行降序排列，然后仅保留 top-k（如 400）个预测框。最

后就是进行 NMS 算法，过滤掉那些重叠度较大的预测框。最后剩余的预测框就是检测结果了。

六. Faster-RCNN 模型简介

与 SSD 算法属于 One Stage 目标检测算法不同的是，Faster-RCNN 属于 Two Stage 算法。Two Stage 目标检测算法先进行区域生成（region proposal, RP）（一个有可能包含待检物体的预选框），再通过卷积神经网络进行样本分类。One Stage 算法则不用进行 RP,直接在网络中提取特征来预测物体分类和位置。

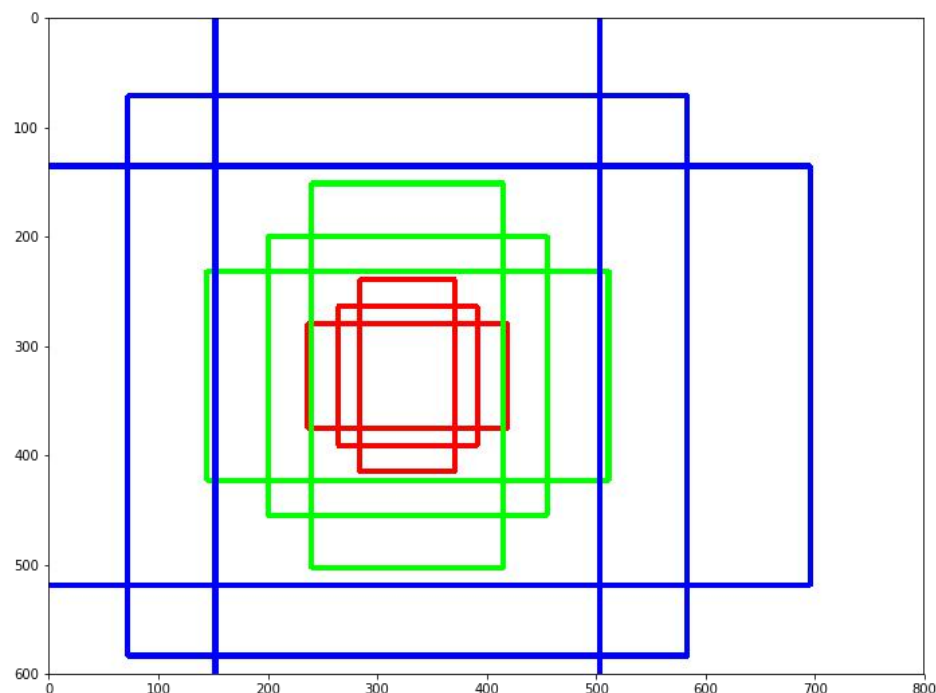
1. Faster-RCNN 结构



Faster-RCNN 是两个网络的结合：用于区域生成(RP)的区域生成网络(RPN, Region Proposal Network)，以及一个用 RPN 生成的区域来检测目标的网络。与 Fast-RCNN 的主要区别也就在于，Fast-CNN 使用 Selective Search 来生成区域建议，而 Faster-RCNN 使用 RPN. RPN 与目标检测网络共享大部分的计算时，RPN 生成区域建议的时间消耗要远小于 Selective Search.简单总结 RPN 就是，RPN 对区域框（叫做 Anchors）进行排序，然后建议出最可能包含目标的那几个。

2. Anchors

Anchors 在 Faster-RCNN 中的角色至关重要。它本质是一个 box，在 Faster-RCNN 的默认配置中，以图像某一像素点为中心的 Anchors 共有 9 个，下图是以(320,320)为中心的 9 个 Anchors:



三种颜色代表着不同的比例(Scales)或大小(Sizes): 128*128, 256*256, 512*512.

以红色为例，红色颜色的框对应着不同的高宽比，分别为:1:1, 1:2 和 2:1.

如果我们以每 16 为步长选择一个位置, 这将会有 1989 (39*51) 个位置. 这将导致 17901 (1989*9) 个框需要考虑. 绝对大小几乎不小于 Sliding Window 和 Pyramid 的结合. 于是我们就可以知道为什么它具有其他 state-of-art 方法一样好的覆盖范围的. 这样的优点是, Faster-RCNN 中的 RP 方法, 能显著减少计算数量.

3. RPN

RPN 的输出是一堆框或者叫区域建议, 用来被后续的分类器和回归器检验, 来最终检查目标是否出现. 更准确地说, RPN 来预测 Anchors 是背景还是前景的可能, 并对 Anchors 进行优化.

- 背景和前景的分类器: 训练分类器的第一步是制作训练数据集. 分类器的基本思想在于给 anchors 一个 label, 将与真实 boxes 与较高重合度的 anchors 标记为前景, 有较低重合度的则标记为背景. 现在 Anchors 都有了自己的 label.
- 边界框(Bouding Box)的回归器: 顺着 label anchors 的过程, 还可以根据相似的标准来挑选 Anchors, 以使回归器优化. 需要注意的是, 被标记为背景的 anchor 不应该被添加到回归器中, 因为它们没有 ground truth. RPN 的总体损失是分类损失和回归损失的组合, 使用的损失函数是:

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^u - v_i), \quad (2)$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \quad (3)$$

4. ROI Pooling

在 RPN 之后，我们得到了具有不同大小的建议区域。不同大小的区域意味着不同大小的 CNN feature map。要构建一个有效的结构来处理不同大小的特征并不容易。感兴趣区域池化(Region of Interest Pooling, ROI)可以通过将 feature maps 缩小为同一大小来简化问题。与固定大小的 Max-Pooling 不同，ROI Pooling 将输入 feature maps 划分为固定数量，例如 k，的大致相等的区域，然后在每个区域上应用 Max-Pooling。因此，无论输入大小如何，ROI Pooling 的输出始终为 k。

5. 四步交替训练

为了强制网络在 RPN 和检测器之间共享 CNN 主干的权重，作者使用了 4 个步骤的训练方法：

- RPN 是独立训练的，这个任务的主干 CNN 是由 ImageNet 分类任务的权重初始化的，之后针对区域建议任务进行微调。
- Faster-RCNN 检测器网络也是独立训练的，这个任务的主干 CNN 也是由那些来自针对 ImageNet 分类任务而训练的网络的权重初始化的，之后针对目标检测任务进行微调。RPN 的权重是固定的，并且由 RPN 生成的建议被用来训练 Faster-RCNN。
- 现在 RPN 被来自 Faster-RCNN 的权重初始化，然后只对区域建议任务进行微调。这次，位于 RPN 和检测器之间的公共层保持固定，只有那些对 RPN 独特的层被微调，这是最终的 RPN。
- Faster-RCNN 检测器再次利用新的 RPN 进行微调。再次，仅对检测器网络唯一的层进行了微调，而公共层权重是固定的。

七. 训练过程

将数据集以 VOC2007 格式呈现之后，均使用 VGG16 作为预训练模型进行训练。修改模型的参数以适应本次作业的数据集。

八. 对于遮挡问题的改进

作业使用 Soft NMS 算法来改进重叠遮挡问题的改进。Soft NMS 对密集物体检测的检测效果有一定的提升作用。

1. NMS

NMS 是在目标检测算法中必备的后续处理步骤，目的是用来去除重复框，也就是降低误检。NMS 算法的过程大概是：首先，根据检测框的分数对它们进行排序，分数最高的框 M 被选择，其它与框 M 重合度很高的框（依据事先定义好的一个法制）则被剔除。这个过程在剩下的框中递归地进行。

那么这种 NMS 算法会导致下图所示的问题：

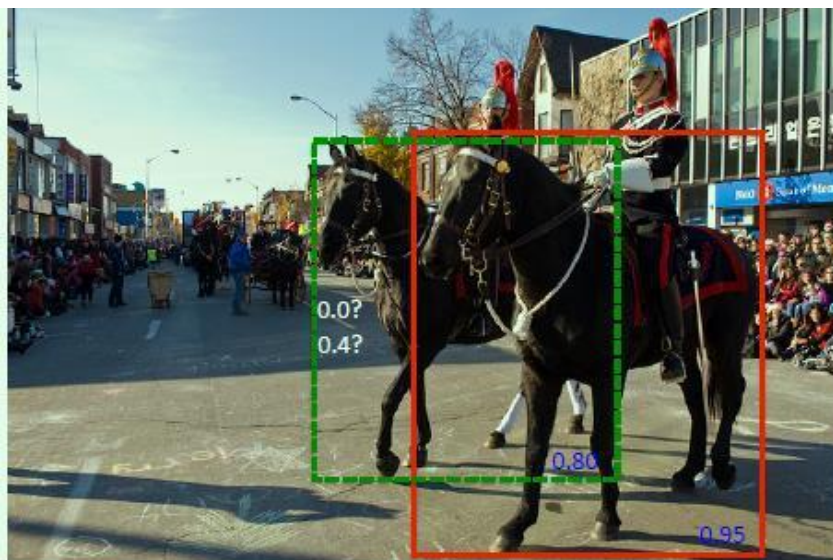


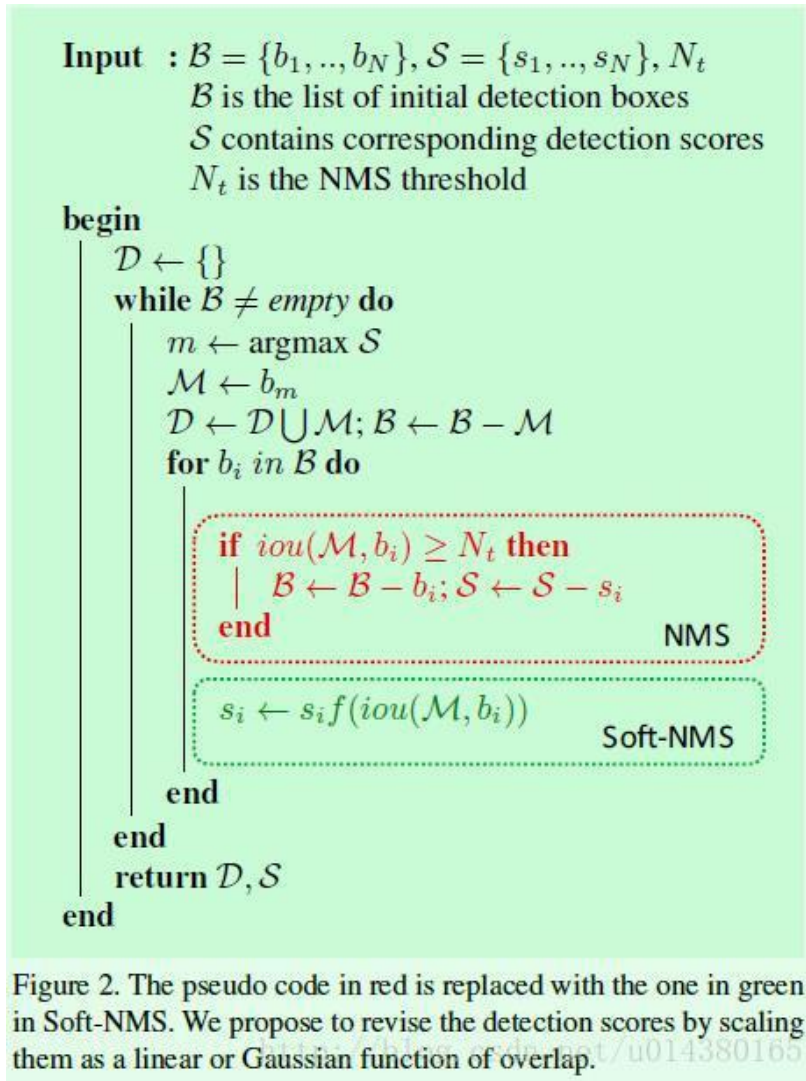
Figure 1. This image has two confident horse detections (shown in red and green) which have a score of 0.95 and 0.8 respectively. The green detection box has a significant overlap with the red one. Is it better to suppress the green box altogether and assign it a score of 0 or a slightly lower score of 0.4?

检测算法本应该输出两个框,但是传统的NMS算法可能会把分数较低的绿框过滤掉(如果绿框和红框的 IOU 大于设定的阈值就会被过滤掉), 导致只检测出一个物体。

2. Soft NMS

可以看出 NMS 的过程有些粗暴, 因为 NMS 直接将与被选择的 box 的 IOU 大于某个阈值的 box 的得分置零, 于是有了 Soft NMS, 该算法简单来说就是用一个稍低一点的分数来代替原有的分数, 而不是直接置零。另外由于 Soft NMS 可以很方便地引入到目标检测算法中, 不需要重新训练原有的模型, 因此这是该算法的一大特点。

Soft NMS 算法过程是: 输入为 B , S , N_t , 含义如下图所示。集合 D 用来放最终的 box, 在集合 B 非空的前提下, 搜索集合 S 中数值最大的数, 假设其下标为 m , 那么 M 就是选择的框。将 M 与 D 集合合并, 再循环集合 B 中的每个 box, 这个时候就体现了与 NMS 的差别, 如果是传统的 NMS 操作, 那么当 B 中的 box b_i 和 M 的 IOU 值大于阈值 N_t , 那么就从 B 和 S 中去除该 box; 如果是 Soft NMS, 则对于 B 中的 box b_i 也是先计算其和 M 的 IOU, 然后该 IOU 值最为函数 $f()$ 的输入, 最后和 b_i 的分数 s_i 相乘作为最后该 b_i 的分数。



接下来的重点就是如何确定函数 $f()$ 了。

对于传统的 NMS 算法可以用下面的式子表示：

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ 0, & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases}$$

为了改进这种粗暴的方法，并遵循 IOU 越大，得分越低的原则，就会想到下面的公式来表示 Soft NMS：

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i (1 - \text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases}$$

但是上面这个公式是不连续的，这样会导致 box 集合中的分数出现断层，于是有下面的 Soft NMS 式子，也是最常用的式子：

$$s_i = s_i e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}, \forall b_i \notin \mathcal{D}$$

这个式子能保证不存在重叠时没有惩罚，存在越高程度的重叠时，惩罚则越高。

九. 实验结果

1. SSD

AP for 带电芯充电宝 = 0.7654
AP for 不带电芯充电宝 = 0.7699
Mean AP = 0.7677
Results: 0.765 0.770 0.768

2. Faster-RCNN

AP for 带电芯充电宝 = 0.8697
AP for 不带电芯充电宝 = 0.8368
Mean AP = 0.8533
Results: 0.870 0.837 0.853

十. 模型对比

- SSD 训练速度更快, 但当速度并不是严格考虑的对象时, Faster-RCNN 要比 SSD 更好, 它的准确率要更高。
- SSD 同样也借鉴了 Faster-RCNN 的 Anchors 技术, 但由于不是在每个位置上的精调更适合实时的处理。同时其准确率要根据实际的应用来判断是否符合准确率的要求。
- 在实验过程中也发现, 图片的分辨率能显著影响模型的准确率, 分辨率降低模型准确率能得到提高。