# RefineDet小结

论文链接：<https://arxiv.org/abs/1711.06897>   
代码链接：<https://github.com/sfzhang15/RefineDet>

|  |  |  |  |
| --- | --- | --- | --- |
| 内容 | 版本 | 作者 | 时间 |

|  |  |  |  |
| --- | --- | --- | --- |
| RefineDet初步总结 | V1.0.0 | 刘畅 | 2018.6.15 |
|  |  |  |  |

目录

[RefineDet小结 1](#_Toc517038554)

[1 概述 2](#_Toc517038555)

[1.1背景 2](#_Toc517038556)

[1.2 理论基础分析 2](#_Toc517038557)

[1.3 算法简介 2](#_Toc517038558)

[2 结构详解 3](#_Toc517038559)

[2.1 ARM 3](#_Toc517038560)

[2.1.1结构 4](#_Toc517038561)

[2.1.2 作用分析 4](#_Toc517038562)

[2.2 TCB和ODM 5](#_Toc517038563)

[2.2.1 结构 5](#_Toc517038564)

[2.2.2 作用分析 5](#_Toc517038565)

[3 实验分析 5](#_Toc517038566)

[3.1 实验一：RefineDet与顶尖目标检测算法对比 6](#_Toc517038567)

[3.2 实验二：ARM、TCB和ODM对检测精度的影响 7](#_Toc517038568)

## 1 概述

### 1.1背景

Single-Shot Refinement Neural Netwrok for Object Detection由中国科学院、中国科学院大学和GE Gloable Research的Shifeng Zhang、Longyin Wen、Xiao Bian、Zhen Lei和Sran Z.Li等人提出，于2018年1月3号上传至ArXiv，收录于CVPR2018.

该论文提出了一种新颖的one-stage的目标检测系统RefineDet，拥有 two-stage方法级别的准确性和one-stage方法的快速性。其思想就是构建一种可以兼顾two-stage方法和one-stage方法优点的检测框架。

### 1.2 理论基础分析

我们都知道，two-stage方法检测的精度高，但是速度慢，通常只有个位数的FPS；但是one-stage方法检测速度快（几十FPS），但是检测精度略低于前者。

要想搭建一个具有two-stage精度的one-stage方法，首先我们要知道为什么前者的精度会比后者高，作者的观点如下：

(1).使用启发式采样的两阶段结构来解决class imbalance问题。

(2).使用联级的两个回归器对边界框进行两次位置修正。

(3).使用两阶段特征来描述目标（如Faster RCNN中RPN之前的first-stage features和RPN后面的second-stage features）。

但是，我认为主要是(1)和(2)决定了两阶段方法精度会高于单阶段方法，(3)的意义不大；此外，两阶段方法大多使用1000×600大小的图片，而单阶段使用300×300到500×500的图片，这对精度的影响也非常巨大，具体可见实验部分。

解释一下，我认为，所谓的class imbalance问题包括两方面：1.正负样本比例不均衡（负样本太多，一般会选择正负比例1:3）；2.样本的难易比例不均衡（简单负样本太多，hard negative太少，模型学习了太多的简单负样本也会导致最终检测效果不好）。针对class imbalance问题，两阶段方法首先会严格限制正负样本比例1:3，对于样本难易度可以使用OHEM方法（详见小结）。

而对于（2）对边界框进行两次位置修正，在两阶段方法中，第一次修正是在RPN中使用卷积层对anchors进行的，第二次修正是在ROI pooling之后进行的。而在单阶段方法中，只对anchors进行了一次位置补偿。我认为这造成了单阶段方法检测结果的IOU比两阶段要低，但是并不会影响召回率，所以单阶段方法检测不到小目标并不是由此造成的。

经过上述分析我们已经知道了两阶段方法为何比单阶段方法精度高，那么如何把两阶段的优点(1)和(2)放到单阶段上呢？

### 1.3 算法简介

作者等人提出了单阶段方法RefineDet，将(1)、(2)的优点融入了进去。RefineDet由三部分组成：ARM（anchor refinement module）、ODM（object detection module）和TCB（transfer connection block）。整体结构如下图所示。

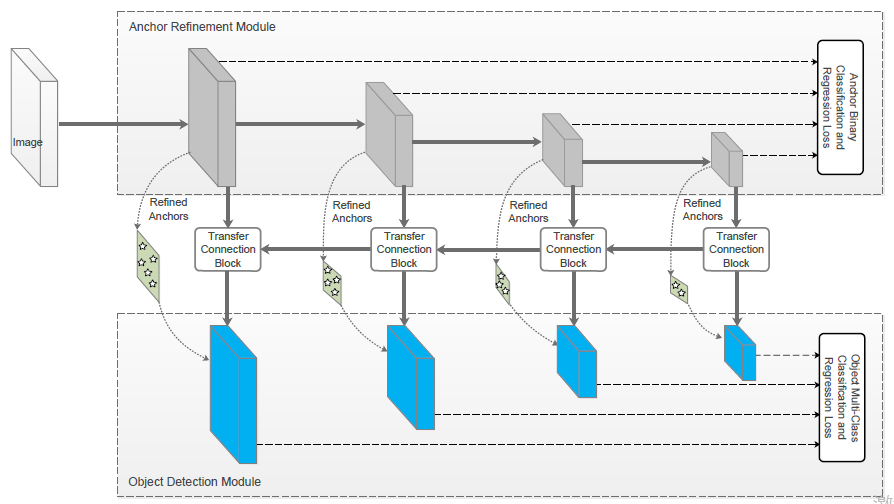


图1 RefineDet整体结构

ARM（anchor refinement module）主要有两个用处：(1).对不同特征图上的anchors进行第一次位置修正和二分类（是否为object），将处理后的anchor做为ODM的输入；整个过程跟RPN非常类似，不过是在多个不同尺度的feature map上进行的。(2).过滤掉过于简单的负样本（Negative Anchor Filtering具体方法见第二部分），为ODM减少搜索空间。

ODM（object detection module）接收ARM的feature map及refine之后的anchors，使用卷积层进一步预测物体的类别及作第二次位置修正。

TCB（transfer connection block）作为桥梁，将ARM中的feature map传给ODM，同时又起到了特征融合的作用，即将高层feature map逐渐通过deconv放大，然后与底层feature map融合，相当于FPN中横向连接的作用。

总的来说，ResfineDet为了获得两阶段检测方法的优点，分别在ARM和ODM中进行分类和位置修正，构建了一个类似两阶段方法的二联级回归器，并且在ARM中使用Negative Anchor Filtering过滤掉简单负样本来解决class imbalance问题。从而，RefineDet拥有与两阶段方法的相媲美的检测精度，同时具备单阶段方法的高效性。

## 2 结构详解

在这一部分中，细致地介绍下ARM、ODM和TCB三个部分。

### 2.1 ARM

### 2.1.1结构

整个ARM部分非常类似SSD的整体结构。以VGG16为基础网络为例，去掉最后的分类层，将fc6和fc7变为卷积层Conv\_fc6（1024×3×3，S=1，P=3）和Conv\_fc7（1024×1×1，S=1，P=0）。然后再增加两个卷积层Conv6\_1（256×1×1，S=1，P=1）和Conv6\_2（512×3×3，S=2，P=1）。

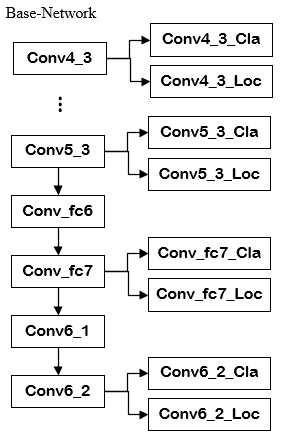
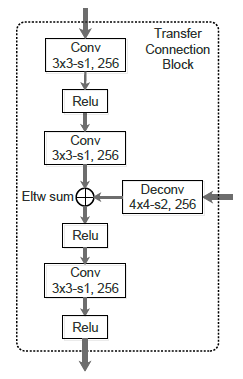
 

图2 ARM结构 图3 TCB结构

分别在Conv4\_3、Conv5\_3、Conv\_fc7和Conv6\_2上对anchors进行二分类和位置补偿。每一层上的anchors只有一种尺度和三种比率（0.5,1,2），即这些feature map的每一个像素点上有三个anchors。所以均使用12×3×3的卷积层预测位置补偿值，用6×3×3的卷积层预测类别。如图2所示。

### 2.1.2 作用分析

ARM（anchor refinement module）主要有两个用处：(1).对不同特征图上的anchors进行第一次位置修正和二分类（是否为object），将处理后的anchor做为ODM的输入；整个过程跟RPN非常类似，不过是在多个不同尺度的feature map上进行的。(2).过滤掉过于简单的负样本（Negative Anchor Filtering），为ODM减少搜索空间。在这里主要解释一下ARM如何用Negative Anchor Filtering来减少搜索空间的。

ARM在feature map的每一个位置都生成3个anchors，卷积后每个位置得到3个refined anchors。这些anchors并不是全部作为ODM的输入进行第二次分类及位置修正，而是要经过Negative Anchor Filtering算法进行处理，过滤掉那些negative confidence大于阈值（0.99）的refined anchors。在训练和前向传播时，这些被过滤掉的anchors（简单负样本）都不会使用，即只用hard negative和positive anchors进行训练。

### 2.2 TCB和ODM

### 2.2.1 结构

TCB（transfer connection block）的结构如图3所示，TCB的结构非常类似FPN的结构，将不同尺度、层次的feature map融合在一起，为ODM（object detection module）提供了更好的特征。

ARM、TCB和ODM的连接情况如下图4所示。TCB作为连接ARM和ODM的桥梁，将融合之后的feature map提供给ODM进行进一步的分类和位置修正。

每一格ODM（如ODM4\_3）都由两个并联的3×3卷积层组成，用于位置修正的Conv通道数为12（3个refine ahchors×4个位置offsets），用于分类的Conv通道数为63（21类×3个refine anchors）。其具体的处理过程与SSD相同，不再赘述。

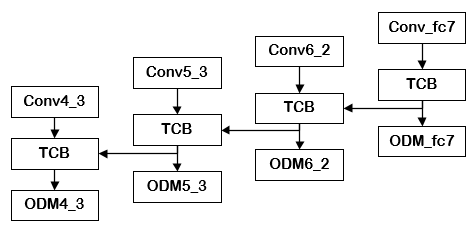


图4 TCB连接ARM和ODM

### 2.2.2 作用分析

TCB有两个作用：(1).对在不同层次、尺度的feature maps进行特征融合，达到类似FPN的效果；(2)将ARM的feature map传给ODM进行进一步的分类和位置修正。至于refined anchors如何传给ODM有待进一步确定。

ODM的唯一作用就是进行二次分类和位置修正，不再赘述。

## 3 实验分析

RefineDet使用多任务代价函数进行end-to-end的训练，batch size32，其他的训练参数依据训练集而定，正负样本的确定方式跟SSD相同，IOU最大及IOU大于0.5的anchors为正样本，数据增强也与SSD相同，详见论文。

在此主要分析两个实验：(1)RefineDet与start-of-the-art检测方法在VOC及COCO上的对比；(2)分析ARM、TCB和ODM对检测精度的影响。其他实验详见论文。

### 3.1 实验一：RefineDet与顶尖目标检测算法对比

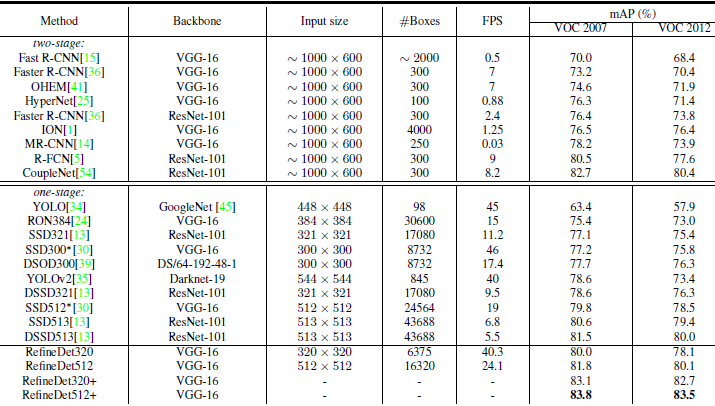


图5 近期VOC数据集各算法检测效果及速度

RefineDet320在40.3FPS的速度下，在VOC2007上mAP为80.0%，在VOC2012上为78.1%。RefineDet512在24.1FPS的速度下，在VOC2007、VOC2012上mAP分别为82.8%和80.1%。

由上图，我总结了如下结论：

(1).两阶段算法的输入图片普遍大于单阶段算法，这使得两阶段算法对小目标的检测效果更好，但是速度更慢（我认为是两阶段算法速度慢的主要原因，大尺寸输入在进行卷积时需要更大的计算量和时间）。

(2).对于单阶段算法来说，随着输入图片的尺寸的增加，mAP显著上升，速度显著下降。如下表所示。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 算法 | 尺寸变化 | | FPS | | mAP（VOC07） | | mAP(VOC2012) | |
| SSD(VGG16) | | | 300->512 | | 46->19 | | 77.2->79.8 | | 75.8->78.5 | |
| SSD(ResNet101) | | | 321->513 | | 11.2->6.8 | | 77.1->80.6 | | 75.4->79.4 | |
| DSSD(ResNet101) | | | 321->513 | | 9.5->5.5 | | 78.6->81.5 | | 76.3->80.0 | |
| RefineDet(VGG16) | | | 320->512 | | 40.3->24.1 | | 80->81.8 | | 78.1->80.1 | |

(3).使用VGG16作为基础网络比ResNet101速度快3倍以上，但是精度的提高根据数据集的不同而变化，在VOC这样比较小的数据集上精度提升一般（1%以内），而在COCO等比较大的数据集上提升显著（3%左右）。可参考下图6.

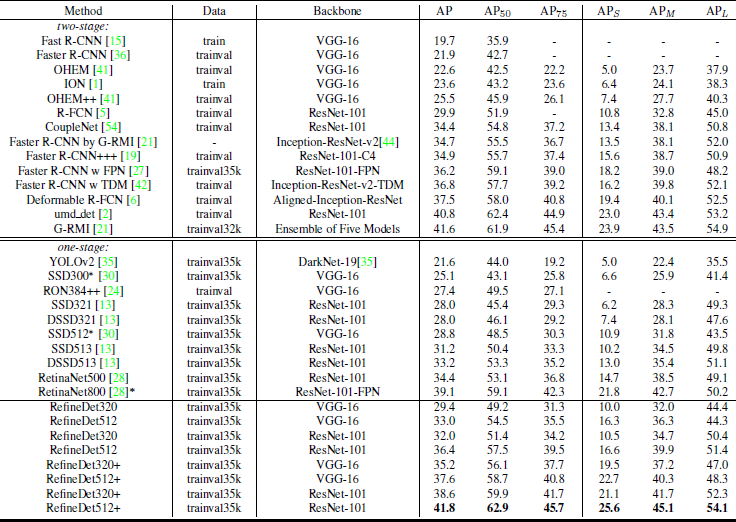


图6 近期COCO数据集各算法检测效果

### 3.2 实验二：ARM、TCB和ODM对检测精度的影响

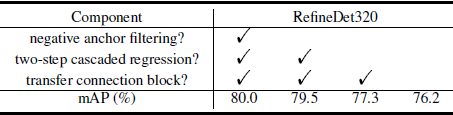


图7 ARM、TCB和ODM对检测精度影响

上述实验使用VOC2007和VOC2012作为训练，在VOC2007测试集上进行测试。使用negtive anchor filtering算法提升了0.5mAP（79.5->80.0），解决了class imbalance问题；使用ODM进行两阶段回归提升了2.2mAP（77.3->79.5）；使用TCB提升了1.1mAP（76.2->77.3）。

总结如下：

(1).使用OMD进行两阶段回归是RefineDet提升检测精度最重要的idea，其思想是通过模仿两阶段算法来获得两阶段算法的优点。但是我认为，两阶段回归可以提升小目标的识别精度和IOU，但是无法提升召回率，并没有解决检测不到小目标的问题。

(2).ARM中negtive anchor filtering算法解决了class imbalance问题，即只使用hard negative和positive样本训练网络，抛弃简单负样本，跟Focal Loss的思想很像。

(3).TCB起到了类似FPN的效果，对浅层大尺度特征增强了语义信息，我认为这是RefineDet提升小目标检测召回率的关键。