Single-Shot Refinement Neural Network for Object Detection

Shifeng Zhang[1,2], Longyin Wen[3], Xiao Bian[3], Zhen Lei[1,2], Stan Z. Li[1,2] [1]
CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China. [2]
University of Chinese Academy of Sciences, Beijing, China. [3]
GE Global Research, Niskayuna, NY.
{shifeng.zhang,zlei,szli}@nlpr.ia.ac.cn, {longyin.wen,xiao.bian}@ge.com

Abstract

*For object detection, the two-stage approach (*e.g.*, Faster R-CNN) has been achieving the highest accuracy, whereas the one-stage approach (*e.g.*, SSD) has the advantage of high efficiency. To inherit the merits of both while overcoming their disadvantages, in this paper, we propose a novel single-shot based detector, called RefineDet, that achieves better accuracy than two-stage methods and maintains comparable efficiency of one-stage methods. RefineDet consists of two inter-connected modules, namely, the anchor refinement module and the object detection module. Specifically, the former aims to (1) filter out negative anchors to reduce search space for the classifier, and (2) coarsely adjust the locations and sizes of anchors to provide better initialization for the subsequent regressor. The latter module takes the refined anchors as the input from the former to further improve the regression and predict multi-class label. Meanwhile, we design a transfer connection block to transfer the features in the anchor refinement module to predict locations, sizes and class labels of objects in the object detection module. The multitask loss function enables us to train the whole network in an end-*

摘要

对目标检测任务（object detection）来说，采用两阶段的方法（比如 Faster R-CNN）已经取得了（目前）最高的准确率，然而采用单一阶段的方法（例如 SSD）拥有很高的效率。为了可以继承两者的优点同时可以克服它们的缺点，在这篇论文中，我们提出了一种新颖的单步（single-shot）检测器，并将其命名为 RefineDet。RefineDet 可以取得比采用两阶段的方法更好的准确率同时也可以保持和单一阶段方法相近的效率。RefineDet 由两个联通的模块组成，即 anchor 细化模块和目标检测模块。前者主要功能是（1）过滤掉负样本 anchor，以便为分类器规约（reduce）搜索空间，以及（2）粗略地调整 anchor 的位置和尺寸，以便可以为接下来的回归器提供更好的初始化结果。后一个模块将经过前一个模块细化之后的 anchor 作为输入，并进一步改善回归过程，同时预测多类别的标签。同时，我们设计了一个传输连接块（transfer connection block）来传输 anchor 细化模块中的特征，以预测目标检测模块中的对象的位置，大小和类别标签。多任务损失函数（the multitask loss function）可以让我们采用端到端（end-to-end）的方式来训练整个网络。在 PASCAL VOC 2007,, PASCAL VOC 2012 和 MS COCO 数据集上的大量实验结果表明：RefineDet 取得了目前最

[1] In case of Faster R-CNN, the features (excluding shared features) in the first stage (*i.e.*, RPN) are trained for the binary classification (being an object or not), while the features (excluding shared features) in the second stage(*i.e.*, Fast R-CNN) are trained for the multi-class classification (background or object classes).

[2] We denote the reference bounding box as “anchor box”, which is also called “anchor” for simplicity, as in [36]. However, in [30], it is called “default box”.

to-end way. Extensive experiments on PASCAL VOC 2007, PASCAL VOC 2012, and MS COCO demonstrate that RefineDet achieves state-of-the-art detection accuracy with high efficiency. Code is available at https://github.com/sfzhang15/RefineDet.

好的（state-of-the-art）检测准确率，同时拥有高的效率。代码可以从这个网址获得：https://github.com/sfzhang15/RefineDet。

1. Introduction

Object detection has achieved significant advances in recent years, with the framework of deep neural networks (DNN). The current DNN detectors of state-of-the-art can be divided into two categories: (1) the two-stage approach, including [3, 15, 36, 41], and (2) the one-stage approach, including [30, 35]. In the two-stage approach, a sparse set of candidate object boxes is first generated, and then they are further classified and regressed. The two-stage methods have been achieving top performances on several challenging benchmarks, including PASCAL VOC [8] and MS COCO [29].

The one-stage approach detects objects by regular and dense sampling over locations, scales and aspect ratios. The main advantage of this is its high computational efficiency. However, its detection accuracy is usually behind that of the two-stage approach, one of the main reasons being due to the class imbalance problem [28].

Some recent methods in the one-stage approach aim to address the class imbalance problem, to improve the detection accuracy. Kong *et al*. [24] use the objectness prior constraint on convolutional feature maps to significantly reduce the search space of objects. Lin *et al*. [28] address the class imbalance issue by reshaping the standard cross entropy loss to focus training on a sparse set of hard examples and down-weights the loss assigned to well-classified examples. Zhang *et al*. [53] design a max-out labeling mechanism to reduce false positives resulting from class imbalance.

In our opinion, the current state-of-the-art two-stage methods, *e.g*., Faster R-CNN [36], R-FCN [5], and FPN [27], have three advantages over the

1、介绍

利用深度神经网络（deep neural network, DNN）框架，目标检测任务近年来取得了重大进展。目前的顶级的 DNN 检测器可以被划分为两类：（1）采用两阶段方法的检测器，包括文献[3, 15, 36, 41]，和（2）采用单一阶段方法的检测器，包括文献[30, 35]。采用两阶段方法的检测器首先会生成一个稀疏集合，里面包括候选的目标框（object boxes），然后折后候选的目标框会被进一步的分类和回归。采用两阶段设计的方法已经在几个具有挑战性的基准上取得了最好的成绩，包括 PASCAL VOC [8]和 MS COCO [29]。

采用单一阶段设计的方法通过有规律的对位置，尺寸以及纵横比进行密集采样来检测物体。这种方法的主要优点是它拥有高计算效率。然而，它们的检测准确率通常不及采用两阶段设计的方法，这其中一个主要的原因是因为类不平衡问题（the class imbalance problem）[28]。

最近，一些采用单一阶段设计的方法力求解决类不平衡问题，以提高检测准确率。Kong 等人[24]在卷积特征图上添加对象先验约束（the objectness prior constraint）来减小对象的搜索空间。Lin 等人[28]通过重新整理（reshape）标准交叉熵损失（the standard cross entropy loss）来将训练过程专注于难例（hard examples）组成的稀疏集合，同时减小已经分类好的样本的损失。Zhang 等人[53]设计了一个最大-出局（max-out）标注机制来减少类不平衡带来的假阳性样本。

在我们看来，目前最好的采用两阶段设计的方法，例如 Faster R-CNN，R-FCN 和 FPN，相比较采用但阶段设计的方法拥有三个优点：（1）使用采样启

one-stage methods as follows: (1) using two-stage structure with sampling heuristics to handle class imbalance; (2) using two-step cascade to regress the object box parameters; (3) using two-stage features to describe the objects[3]. In this work, we design a novel object detection framework, called RefineDet, to inherit the merits of the two approaches (*i.e.*, one-stage and two-stage approaches) and overcome their shortcomings. It improves the architecture of the one-stage approach, by using two inter-connected modules (see Figure 1), namely, the anchor [4]refinement module (ARM) and the object detection module (ODM). Specifically, the ARM is designed to (1) identify and remove negative anchors to reduce search space for the classifier, and (2) coarsely adjust the locations and sizes of anchors to provide better initialization for the subsequent regressor. The ODM takes the refined anchors as the input from the former to further improve the regression and predict multi-class labels. As shown in Figure 1, these two inter-connected modules imitate the two-stage structure and thus inherit the three aforementioned advantages to produce accurate detection results with high efficiency. In addition, we design a transfer connection block (TCB) to transfer the features[5] in the ARM to predict locations, sizes, and class labels of objects in the ODM. The multi-task loss function enables us to train the whole network in an end-to-end way.

发式的两阶段结构来处理类别不平衡问题；（2）使用两步级联来回归对象框参数；（3）使用两阶段特征来描述对象。在这里，我们设计了一个新颖的目标检测框架，称之为 RefineDet，RefineDet 继承了两种方法（例如，单一阶段设计方法，两阶段设计方法）的优点，并克服了它们的缺点。它改进了采用单一阶段设计的方法，使用两个相连的模块（见图 1），即 anchor 细化模块（the anchor refinement module，ARM）和目标检测模块（the object detection module，ODM）。具体而言，ARM 主要是为了（1）确定和移除负样本 anchor 以为分类器减少搜索空间，（2）粗略地调整 anchor 的位置和尺寸，以便可以为接下来的回归器提供更好的初始化结果。ODM 将前者提供的经过细化后的 anchors 作为输入，进一步改善回归过程，预测多类别标签。如图 1 所示，这两个相连的模块模仿了采用两阶段设计的方法，因此继承了上述的三个优点，可以以高效率计算准确的检测结果。另外，我们设计了一个传输连接块（transfer connection block，TCB）来传输 anchor 细化模块中的特征，一边可以在 ODM 中预测目标的位置，尺寸和类别标签。多任务损失函数可以让我们采用端到端的方式来训练整个网络。

[3] In case of Faster R-CNN, the features (excluding shared features) in the first stage (i.e., RPN) are trained for the binary classification (being an object or not), while the features (excluding shared features) in the second stage (i.e., Fast R-CNN) are trained for the multi-class classification (background or object classes).

[4] We denote the reference bounding box as “anchor box”, which is also called “anchor” for simplicity, as in [36]. However, in [30], it is called “default box”.

[5] The features in the ARM focus on distinguishing positive anchors from background. We design the TCB to transfer the features in the ARM to handle the more challenging tasks in the ODM, *i.e.*, predict accurate object locations, sizes and multi-class labels.

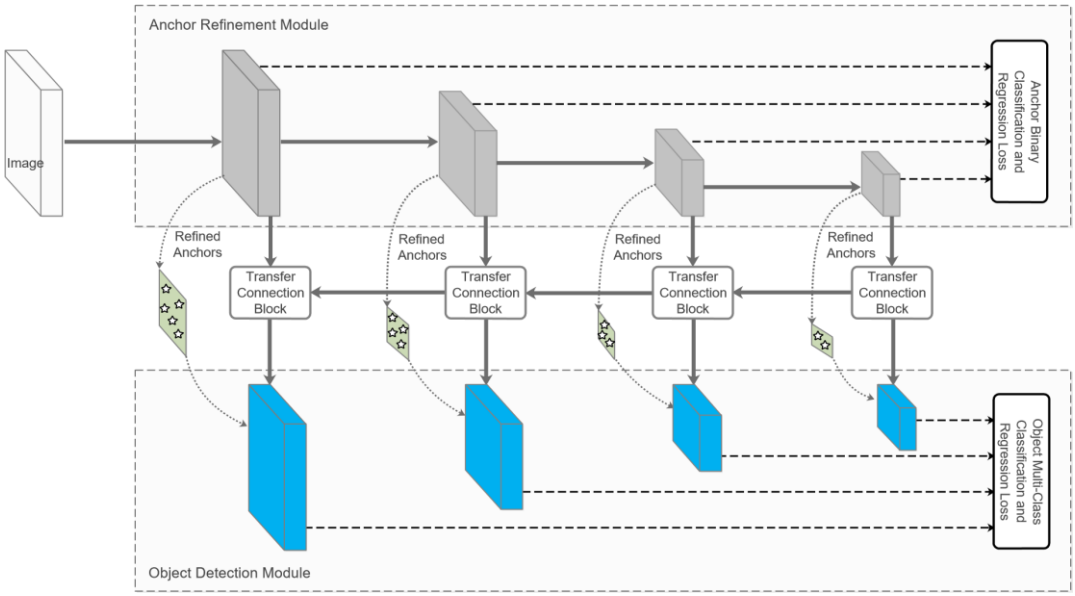

Figure 1: Architecture of RefineDet. For better visualization, we only display the layers used for detection. The celadon parallelograms denote the refined anchors associated with different feature layers. The stars represent the centers of the refined anchor boxes, which are not regularly paved on the image.

图 1：RefineDet 网络结构，为了可以更好的可视化，我们仅仅展示了检测所需要使用的层。灰绿色菱形代表和不同的特征层相联系的经过细化的 anchors。星形代表已定义的 anchors 的中心，这些 anchors 在图像上并不按照一定规则铺设。

Extensive experiments on PASCAL VOC 2007, PASCAL VOC 2012, and MS COCO benchmarks demonstrate that RefineDet outperforms the state-of-the-art methods. Specifically, it achieves 85.8% and 86.8% mAPs on VOC 2007 and 2012, with VGG-16 network. Meanwhile, it outperforms the previously best published results from both one-stage and two-stage approaches by achieving 41.8% AP[6] on MS COCO test-dev with ResNet-101. In addition, RefineDet is time efficient, *i.e.*, it runs at 40.2 FPS and 24.1 FPS on a NVIDIA Titan X GPU with the input sizes 320 × 320 and 512 × 512 in inference.

The main contributions of this work are summarized as follows. (1) We introduce a novel one-stage framework for object detection, composed of two inter-connected modules, *i.e.*, the ARM and the ODM. This leads to performance better than the two-stage approach while

在 PASCAL VOC 2007，PASCAL VOC 2012 和 MS COCO 数据集基准上的大量实验结果表明：RefineDet 比目前最好的方法表现更好。具体地，它使用 VGG-16 网络在 VOC2007 和 2012 数据集上分别达到了 85.8% and 86.8% mAPs。同时，它通过使用 ResNet-101 在 MS COCO 测试开发中获得 41.8%的 AP，胜过了之前公布的一阶段和两阶段方法的最佳结果。另外，RefineDet 在时间上也是十分高效，例如，在 NVIDIA Titan X GPU 上，输入尺寸分别为 320 × 320 和 512 × 512 的情况下，它可以分别以 40.2 FPS 和 24.1 FPS 的速率运行。

本文的主要贡献可以总结如下：（1）我们介绍了一种新颖的但一阶段目标检测框架，该框架由两个相连的模块组成，分别是 ARM 模块，ODM 模块。这导致本网络性能比采用两阶段设计的方法更好，同时保持单阶段方法的高效率。（2）为了保证高效性，我们设计了 TCB 用来传输 ARM 中的特征，以

[6] Based on the evaluation protocol in MS COCO [29], AP is the single most important metric, which is computed by averaging over all 10 intersection over union (IoU) thresholds (i.e., in the range [0.5:0.95] with uniform step size 0.05) of 80 categories.

maintaining high efficiency of the one-stage approach. (2) To ensure the effectiveness, we design the TCB to transfer the features in the ARM to handle more challenging tasks, *i.e.*, predict accurate object locations, sizes and class labels, in the ODM. (3) RefineDet achieves the latest state-of-the-art results on generic object detection (*i.e.*, PASCAL VOC 2007 [10], PASCAL VOC 2012 [11] and MS COCO [29]).

解决更有挑战性的任务，如早 ODM 中准确预测目标的位置，尺寸和类别标签。(3) RefineDet 取得了在通用目标检测(如 PASCAL VOC 2007 [10]，PASCAL VOC 2012 [11] 和 MS COCO [29]) 方面目前最好的结果。

2. Related Work

Classical Object Detectors. Early object detection methods are based on the sliding-window paradigm, which apply the hand-crafted features and classifiers on dense image grids to find objects. As one of the most successful methods, Viola and Jones [47] use Haar feature and AdaBoost to train a series of cascaded classifiers for face detection, achieving satisfactory accuracy with high efficiency. DPM [12] is another popular method using mixtures of multiscale deformable part models to represent highly variable object classes, maintaining top results on PASCAL VOC [8] for many years. However, with the arrival of deep convolutional network, the object detection task is quickly dominated by the CNN-based detectors, which can be roughly divided into two categories, *i.e.*, the two-stage approach and one-stage approach.

Two-Stage Approach. The two-stage approach consists of two parts, where the first one (*e.g.*, Selective Search [46], EdgeBoxes [55], DeepMask [32, 33], RPN [36]) generates a sparse set of candidate object proposals, and the second one determines the accurate object regions and the corresponding class labels using convolutional networks. Notably, the two-stage approach (*e.g.*, R-CNN [16], SPPnet [18], Fast RCNN [15] to Faster R-CNN [36]) achieves dominated performance on several challenging datasets (*e.g.*, PASCAL VOC 2012 [11] and MS COCO [29]). After that, numerous effective techniques are proposed to further improve the performance, such as architecture

2、相关工作

经典目标检测器。早期的目标检测方法基于滑动窗口的规范，它们在密集图像网络上医用手动提取的特征和分类器来寻找目标。作为最成功的方法之一，Viola 和 Jones [47] 使用 Haar 特征和 Adaboost 来训练一系列用于然连检测的级联分类器，以高效率达到了令人满意的精度。DPM[12]是另一种流行的方法，它使用多尺度可变性零件模型的混合来表示高度可变的目标类别，多年来在 PASCAL VOC [8]数据集上保持了最佳结果。然而，随着深度卷积网络的到来，目标检测任务迅速被基于 CNN 的检测器所支配，其大致可以分为两类，即采用两阶段设计的方法和采用一阶段设计的方法。

两阶段设计方法。采用两阶段设计的方法由两部分组成，其中第一部分（如：Selective Search [46], EdgeBoxes [55], DeepMask [32, 33], RPN [36]）生成一个包含候选目标 proposals 的稀疏集合；第二部分使用卷积网络来确定准确的目标区域和相应的类别标签。值得注意的是，采用两阶段设计的方法（如：R-CNN [16], SPPnet [18], Fast RCNN [15] 到 Faster R-CNN [36]）在几个具有挑战性的数据集上由卓越的性能表现（如：PASCAL VOC 2012 [11] 和 MS COCO [29]）。之后，提出了寻多有效的技术来进一步提高性能，比如框架图（architecture diagram [5, 26, 54]），训练策略（training strategy [41, 48]），上下文推理（contextual reasoning [1, 14, 40, 50]）

diagram [5, 26, 54], training strategy [41, 48], contextual reasoning [1, 14, 40, 50] and multiple layers exploiting [3, 25, 27, 42].

One-Stage Approach. Considering the high efficiency, the one-stage approach attracts much more attention recently. Sermanet *et al*. [38] present the OverFeat method for classification, localization and detection based on deep ConvNets, which is trained end-to-end, from raw pixels to ultimate categories. Redmon *et al*. [34] use a single feedforward convolutional network to directly predict object classes and locations, called YOLO, which is extremely fast. After that, YOLOv2 [35] is proposed to improve YOLO in several aspects, *i.e*., add batch normalization on all convolution layers, use high resolution classifier, use convolution layers with anchor boxes to predict bounding boxes instead of the fully connected layers, etc. Liu *et al*. [30] propose the SSD method, which spreads out anchors of different scales to multiple layers within a ConvNet and enforces each layer to focus on predicting objects of a certain scale. DSSD [13] introduces additional context into SSD via deconvolution to improve the accuracy. DSOD [39] designs an efficient framework and a set of principles to learn object detectors from scratch, following the network structure of SSD. To improve the accuracy, some one-stage methods [24, 28, 53] aim to address the extreme class imbalance problem by re-designing the loss function or classification strategies. Although the one-stage detectors have made good progress, their accuracy still trails that of two-stage methods.

3. Network Architecture

Refer to the overall network architecture shown in Figure 1. Similar to SSD [30], RefineDet is based on a feedforward convolutional network that produces a fixed number of bounding boxes and the scores indicating the presence of different classes of objects in those boxes, followed by the non-maximum suppression to produce the final result. RefineDet is formed by two inter-connected

以及多层次开发（multiple layers exploiting [3, 25, 27, 42]）。

单阶段设计方法。考虑到高效率的实现，最近单阶段设计的方法收到了很多的关注。Sermanet 等人[38]提出了基于 ConvNets 的分类，定位和检测的 OverFeat 方法，该方法采用从原始像素信息到最终类别的端到端方法训练。Redmon 等人[34]使用单一的前馈卷积网络直接预测目标类别和位置，该方法被称之为 YOLO，运行快速。之后 YOLOv2 [35]被提出，YOLOv2 在几个方面对 YOLO 进行了改进，如在所有的卷积层上都加入了批归一化操作（batch normalization），使用高分辨率的分类器，使用带 anchor boxes 的卷积层而不是全连接层来预测开宇边界框等等。Liu 等人[30]提出了 SSD 方法，它将不同尺度的 anchors 分散到 ConvNet 中的不同层，并强制每个层专注于预测特定尺度的目标。DSSD [13]通过逆卷积（deconvolution）过程将额外的上下文信息引入 SSD 来提高准确率。DSOD [39]遵循 SSD 的网络架构，设计了一个搞笑的框架和一套原理，从零开始学习一个目标检测器。为了提高准确性，一些采用单阶段设计的方法[24, 28, 53]旨在通过重新设计损失函数和分类策略来解决极端类别失衡问题。尽管采用单阶段设计的检测器已经取得了很好的进展，但是它们的准确性仍然落后于采用两阶段设计的方法。

3、网络结构

参见图 1 所示的总体网络架构。与 SSD[30]相似，RefineDet 基于前馈卷积网络，该网络生成固定数目的边界框和表示在这些框中存在的不同类别目标的分数，通过非最大抑制（the non-maximum suppression）来产生最后的结果。RefineDet 由两个相连的模块组成，即 ARM 和 ODM。ARM 旨在移除负样本 anchors 以便为分类器减少搜索空间，并粗略调整 anchors 的位置和大小，以便为随后的

modules, *i.e.*, the ARM and the ODM. The ARM aims to remove negative anchors so as to reduce search space for the classifier and also coarsely adjust the locations and sizes of anchors to provide better initialization for the subsequent regressor, whereas ODM aims to regress accurate object locations and predict multi-class labels based on the refined anchors. The ARM is constructed by removing the classification layers and adding some auxiliary structures of two base networks (*i.e.*, VGG-16 [43] and ResNet-101 [19] pretrained on ImageNet [37]) to meet our needs. The ODM is composed of the outputs of TCBs followed by the prediction layers (*i.e.*, the convolution layers with 3×3 kernel size), which generates the scores for object classes and shape offsets relative to the refined anchor box coordinates. The following explain three core components in RefineDet, *i.e.*, (1) transfer connection block (TCB), converting the features from the ARM to the ODM for detection; (2) two-step cascaded regression, accurately regressing the locations and sizes of objects; (3) negative anchor filtering, early rejecting well-classified negative anchors and mitigate the imbalance issue.

Transfer Connection Block. To link between the ARM and ODM, we introduce the TCBs to convert features of different layers from the ARM, into the form required by the ODM, so that the ODM can share features from the ARM. Notably, from the ARM, we only use the TCBs on the feature maps associated with anchors. Another function of the TCBs is to integrate large-scale context [13, 27] by adding the high-level features to the transferred features to improve detection accuracy. To match the dimensions between them, we use the deconvolution operation to enlarge the high-level feature maps and sum them in the element-wise way. Then, we add a convolution layer after the summation to ensure the discriminability of features for detection. The architecture of the TCB is shown in Figure 2.

回归器提供更好的初始化结果。而 ODM 旨在根据细化后的 anchors 将结果回归到准确的目标位置并预测多类别标签。通过去除分类层并添加两个基础网络（即 VGG-16 [43] 和 ResNet-101 [19]，事先在 ImageNet [37]数据集上训练完成）的一些辅助结构，来构建 ARM，以满足我们的需求。ODM 由 TCB 的输出组成，TCB 后面连接着预测层，其生成目标类别的分数和相对于细化后的 anchors 的坐标的形状偏移量。下面解释 RefineDet 中的三个核心组件，即（1）传输连接块（TCB），将特征从 ARM 转换到 ODM 以供检测；（2）两部级联回归，准确地回归目标的位置和大小；（3）负样本 anchor 过滤，尽早拒绝已经分类好的负样本 anchors，环节失衡问题。

传输连接块。为了在 ARM 和 ODM 之间建立链接，我们引入了 TCB，将来自 ARM 的不同层的功能转换为 ODM 所需的形式，以便 ODM 可以共享来自 ARM 的特征。值得注意的是，我们旨在与 anchors 相关联的特征图上使用 TCB。TCB 的另一个功能是通过将高级特征添加到传输的特征来继承大规模的上下文[13, 27]，以提高检测的准确性。为了让他们之间的维度相匹配，我们使用逆卷积操作来增大高级特征图，并把它们的对应元素进行求和。然后，我们在求和之后添加卷积层以确保检测的特征的可辨性。TCB 的体系结构如图 2 所示。

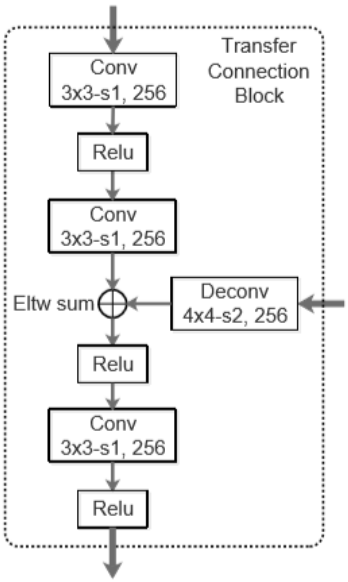

Figure 2 The overview of the transfer connection block.

图 2 传输连接层概述。

Two-Step Cascaded Regression. Current one-stage methods [13, 24, 30] rely on one-step regression based on various feature layers with different scales to predict the locations and sizes of objects, which is rather inaccurate in some challenging scenarios, especially for the small objects. To that end, we present a two-step cascaded regression strategy to regress the locations and sizes of objects. That is, we use the ARM to first adjust the locations and sizes of anchors to provide better initialization for the regression in the ODM. Specifically, we associate n anchor boxes with each regularly divided cell on the feature map. The initial position of each anchor box relative to its corresponding cell is fixed. At each feature map cell, we predict four offsets of the refined anchor boxes relative to the original tiled anchors and two confidence scores indicating the presence of foreground objects in those boxes. Thus, we can yield n refined anchor boxes at each feature map cell.

After obtaining the refined anchor boxes, we pass them to the corresponding feature maps in the ODM to further generate object categories and accurate object locations and sizes, as shown in

两步串联回归。目前的采用单阶段设计的方法[13, 24, 30]依赖于基于不同尺度下的各个特征层的单步回归来预测目标的位置和大小，这在一些具有挑战性的场景中是非常不准确的，特别是对于小目标而言。为此，我们提出了一个两步级联回归策略来回归目标的位置和大小。也就是说，我们使用 ARM 来首次调整 anchors 的位置和大小，以便为 ODM 中的回归操作提供更好的初始化结果。具体而言，我们将 n 个 anchor boxes 与特定特征图上的每个规则划分的单元相互关联。每个 anchor box 相对于其对应单元的初始位置是固定的。对于每个特征图单元，我们预测预测经过细化的 anchor boxes 相对于原始平铺 anchors 的四个偏移量以及便是这些框中存在前景对象的两个置信度分数。因此，我们可以在每个特征图单元中生成 n 个细化后的 anchor boxes。

如图 1 所示，在获得细化后的 anchor boxes 之后，我们将它们传递给 ODM 中相应的特征图，以进一步生成目标类别和准确的目标位置和大小。ARM 和 ODM 中相对应的特征图具有相同的维度。

Figure 1. The corresponding feature maps in the ARM and the ODM have the same dimension. We calculate c class scores and the four accurate offsets of objects relative to the refined anchor boxes, yielding $c + 4$ outputs for each refined anchor boxes to complete the detection task. This process is similar to the default boxes used in SSD [30]. However, in contrast to SSD [30] directly uses the regularly tiled default boxes for detection, RefineDet uses two-step strategy, *i.e.*, the ARM generates the refined anchor boxes, and the ODM takes the refined anchor boxes as input for further detection, leading to more accurate detection results, especially for the small objects.

Negative Anchor Filtering. To early reject well-classified negative anchors and mitigate the imbalance issue, we design a negative anchor filtering mechanism. Specifically, in training phase, for a refined anchor box, if its negative confidence is larger than a preset threshold θ (*i.e.*, set $\theta = 0.99$ empirically), we will discard it in training the ODM. That is, we only pass the refined hard negative anchor boxes and refined positive anchor boxes to train the ODM. Meanwhile, in the inference phase, if a refined anchor box is assigned with a negative confidence larger than θ, it will be discarded in the ODM for detection.

4. Training and Inference

Data Augmentation. We use several data augmentation strategies presented in [30] to construct a robust model to adapt to variations of objects. That is, we randomly expand and crop the original training images with additional random photometric distortion [20] and flipping to generate the training samples. Please refer to [30] for more details.

Backbone Network. We use VGG-16 [43] and ResNet-101 [19] as the backbone networks in our RefineDet, which are pretrained on the ILSVRC CLS-LOC dataset [37]. Notably, RefineDet can also work on other pretrained networks, such as Inception V2 [22], Inception ResNet [44], and ResNeXt101 [49]. Similar to DeepLab-LargeFOV [4], we convert fc6

我们计算 c 个类别分数和相对于细化 anchor boxes 的四个精确的偏移量，为每个细化 anchor box 产生 c+4 个输出以实现检测任务。该过程与 SSD[30]中使用的默认框（default boxes）类似。然而，和 SSD[30]直接使用规则平铺的默认框进行检测相比，RefineDet 使用两步策略，即 ARM 生成细化后的 anchor boxes，ODM 将这些细化后的 anchor boxes 作为输入用于进一步检测，这样会使得检测结果更加精确，特别是对小目标而言。

负样本 anchor 过滤。为了尽早拒绝已经被很好的分类的负样本 anchors，并缓解失衡问题，我们设计了一个负样本 anchors 过滤机制。具体而言，在训练阶段，对于一个细化后的 anchor box，如果其负置信度大于一个预先设置好的阈值θ（如，一般经验性地设置θ=0.99），我们会在训练 ODM 的过程中舍弃该 anchor。即，我们只传递经过细化后的负样本 anchor boxes 和正样本 anchor boxes 来训练 ODM。同时，在推理阶段，如果一个一个经过细化后的 anchor box 被赋予一个大于θ的负置信度，那么该 anchor 将在 ODM 的检测过程中被舍弃。

4、训练和推理

数据增强。我们使用在参考文献[30]中提到的几种数据增强策略来构建一个适应对象变化的健壮（robust）模型。即，我们随机扩展和裁剪原始训练图像，并附加随机光度测量失真[20]并翻转生成训练样本。 请参阅[30]了解更多具体细节。

主干网络。在 RefineDet 中，我们使用 VGG-16 [43] 和 ResNet-101 [19]作为主干网络，它们预先在 ILSVRC CLS-LOC 数据集[37]上进行训练。值得注意的是，RefineDet 也可以使用其他的预训练网络，比如 Inception V2 [22], Inception ResNet [44]，和 ResNeXt101 [49]。和 DeepLab-LargeFOV [4]类似，我们通过子采样（subsample）参数将 VGG-16 的 fc6

and fc7 of VGG-16 to convolution layers conv fc6 and conv fc7 via subsampling parameters. Since conv4 3 and conv5 3 have different feature scales compared to other layers, we use L2 normalization [31] to scale the feature norms in conv4 3 and conv5 3 to 10 and 8, then learn the scales during back propagation. Meanwhile, to capture high-level information and drive object detection at multiple scales, we also add two extra convolution layers (*i.e.*, conv6 1 and conv6 2) to the end of the truncated VGG-16 and one extra residual block (*i.e.*, res6) to the end of the truncated ResNet101, respectively.

Anchors Design and Matching. To handle different scales of objects, we select four feature layers with the total stride sizes 8, 16, 32, and 64 pixels for both VGG-16 and ResNet101[7], associated with several different scales of anchors for prediction. Each feature layer is associated with one specific scale of anchors (*i.e.*, the scale is 4 times of the total stride size of the corresponding layer) and three aspect ratios (*i.e.*, 0.5, 1.0, and 2.0). We follow the design of anchor scales over different layers in [53], which ensures that different scales of anchors have the same tiling density [51, 52] on the image. Meanwhile, during the training phase, we determine the correspondence between the anchors and ground truth boxes based on the jaccard overlap [7], and train the whole network end-to-end accordingly. Specifically, we first match each ground truth to the anchor box with the best overlap score, and then match the anchor boxes to any ground truth with overlap higher than 0.5.

Hard Negative Mining. After matching step, most of the anchor boxes are negatives, even for the ODM, where some easy negative anchors are rejected by the ARM. Similar to SSD [30], we use hard negative mining to mitigate the extreme foreground-background class imbalance, *i.e.*, we select some negative anchor boxes with top loss values to make the ratio between the negatives and

和 fc7 转换为卷积层 conv fc6 和 conv fc7。由于 conv4 3 和 conv5 3 与其他图层具有不同的特征尺度（feature scales），因此我们使用 L2 归一化（L2 normalization）[31]将 conv4 3 和 conv5 3 中的特征尺度缩放到 10 和 8，然后在后向传播期间学习尺度。同时，为了在多个尺度上捕获高级信息和驱动目标检测，我们分别还在截断的 VGG-16 模型的尾部增加了两个额外的卷积层（即 conv6 1 和 conv6 2），在截断的 ResNet101 模型尾部额外增加了一个残差快（即，res6）。

Anchors 设计与匹配。为了处理不同尺度的目标，我们选择了四个特征层，其中 VGG-16 和 ResNet101 的总步长大小分别为 8，16，32 和 64 像素，与预测的几种不同尺度的 anchors 相关联。每个特征层都与一个特定的 anchors 的尺度（即尺度是相应层总步长的 4 倍）和三个纵横比（即 0.5，1.0，2.0）相关联。我们遵循参考文献[53]中不同层次 anchor 尺度的设计，这保证了不同 anchors 的尺度具有相同的图片平铺密度[51, 52]。同时，在训练阶段，我们根据 jaccard 重叠[7]决定 anchors 和真是框之间的对应关心，并相应地对整个网络进行端到端地训练。具体地，我们首先将每个真实框与和其具有最佳重叠分数地 anchor boxes 进行匹配，然后将 anchor boxes 与和其重叠度高于 0.5 的任何真实框进行匹配。

难分样本挖掘（Hard Negative Mining）。在匹配之后，大多数 anchor boxes 都属于负样本，即使对于 ODM，此时一些容易识别的负样本已经被 ARM 拒绝。和 SSD 类似，我们使用难分样本挖掘来缓解一段的前景-背景失衡问题，即在训练时我们没有使用所有的负样本 anchors 或者随机选择负样本 anchors，而是选择一些具有最高损失值的负

[7] For the VGG-16 base network, the conv4 3, conv5 3, conv fc7, and conv6 2 feature layers are used to predict the locations, sizes and confidences of objects. While for the ResNet-101 base network, res3b3, res4b22, res5c, and res6 are used for prediction.

positives below 3 : 1, instead of using all negative anchors or randomly selecting the negative anchors in training.

Loss Function. The loss function for RefineDet consists of two parts, *i.e.*, the loss in the ARM and the loss in the ODM.

For the ARM, we assign a binary class label (of being an object or not) to each anchor and regress its location and size simultaneously to get the refined anchor. After that, we pass the refined anchors with the negative confidence less than the threshold to the ODM to further predict object categories and accurate object locations and sizes. With these definitions, we define the loss function as:

样本 anchors，以保证负样本和正样本的比率低于 3：1，

损失函数。RefineDet 的损失函数由两部分组成，即 ARM 的损失和 ODM 的损失。

对于 ARM，我们为每个 anchor 赋予一个二元类标签（表明是一个目标或者不是一个目标），并且同时回归它的位置和大小，以获得一个细化的 anchor。之后，我们将具有小于阈值负置信度的细化后的 anchors 传递给 ODM 以便进一步预测目标类别和准确的目标位置和大小。我们将损失函数定义如下：

$$\begin{aligned}\mathcal{L}(\{p_i\},\{x_i\},\{c_i\},\{t_i\}) = &\frac{1}{N_{\text{arm}}}\Big(\sum_i \mathcal{L}_{\text{b}}(p_i,[l_i^* \geq 1]) \\ &+\sum_i [l_i^* \geq 1]\mathcal{L}_{\text{r}}(x_i, g_i^*)\Big) + \frac{1}{N_{\text{odm}}}\Big(\sum_i \mathcal{L}_{\text{m}}(c_i, l_i^*) \\ &+\sum_i [l_i^* \geq 1]\mathcal{L}_{\text{r}}(t_i, g_i^*)\Big)\end{aligned} \tag{1}$$

where i is the index of anchor in a mini-batch, l_i^* is the ground truth class label of anchor i, g_i^* is the ground truth location and size of anchor i. p_i and x_i are the predicted confidence of the anchor i being an object and refined coordinates of the anchor i in the ARM. c_i and t_i are the predicted object class and coordinates of the bounding box in the ODM. N_{arm} and N_{odm} are the numbers of positive anchors in the ARM and ODM, respectively. The binary classification loss L_b is the cross-entropy/log loss over two classes (object *vs.* not object), and the multi-class classification loss L_m is the softmax loss over multiple classes confidences. Similar to Fast R-CNN [15], we use the smooth L1 loss as the regression loss L_r. The Iverson bracket indicator function $[l_i^* \geq 1]$ outputs 1 when the condition is true, *i.e.*, $l_i^* \geq 1$ (the anchor is not the negative), and 0 otherwise. Hence $[l_i^* \geq 1]\text{L}_\text{r}$ indicates that the regression loss is ignored for negative anchors. Notably, if $N_{\text{arm}} = 0$, we set $\text{L}_\text{b}(p_i, [l_i^* \geq 1]) = 0$ and $\text{L}_\text{r}(x_i, g_i^*) = 0$; and if $N_{\text{odm}} = 0$, we set $\text{L}_\text{m}(c_i, l_i^*) = 0$ and $\text{L}_\text{r}(t_i, g_i^*) = 0$ accordingly.

这里 i 代表一个小批次里面 anchor 的索引，l_i^*是 anchor i 的真实类别标签，g_i^*是 anchor i 真是的位置和大小。p_i 和 x_i 分别表示预测的 anchor i 是一个目标的置信程度和 ARM 中细化后的 anchor i 的坐标。c_i 和 t_i 分别表示 ODM 中预测的边界框的物体类别和坐标。N_{arm} 和 N_{odm} 分别指 ARM 和 ODM 中正样本 anchors 的数目。二分类损失 L_b 是两个类别的交叉熵/log 损失（是一个目标或者不是一个目标），多类别分类损失 L_m 是多个类别置信度的 softmax 损失。和 Fast R-CNN [15]类似，我们使用平滑 L1 损失（the smooth L1 loss）作为回归损失 L_r。Iverson 括号指示函数（the Iverson bracket indicator function）$[l_i^* \geq 1]$输出 1 当条件为真时，即$l_i^* \geq 1$（anchor 不是负样本时），反之则输出 0。因此$[l_i^* \geq 1]\text{L}_\text{r}$ 表明负样本 anchors 的回归损失被忽略。值得注意的是，如果 $N_{\text{arm}} = 0$，我们相对应地设置 $\text{L}_\text{b}(p_i, [l_i^* \geq 1]) = 0$ 和 $\text{L}_\text{r}(x_i, g_i^*) = 0$；当 $N_{\text{odm}} = 0$ 时，我们相对应地设置 $\text{L}_\text{m}(c_i, l_i^*) = 0$ 和 $\text{L}_\text{r}(t_i, g_i^*) = 0$。

Optimization. As mentioned above, the backbone network (*e.g.*, VGG-16 and ResNet-101) in our RefineDet method is pretrained on the ILSVRC CLS-LOC dataset [37]. We use the "xavier" method [17] to randomly initialize the parameters in the two extra added convolution layers (*i.e.*, conv6 1 and conv6 2) of VGG-16 based RefineDet, and draw the parameters from a zero-mean Gaussian distribution with standard deviation 0.01 for the extra residual block (*i.e.*, res6) of ResNet-101 based RefineDet. We set the default batch size to 32 in training. Then, the whole network is fine-tuned using SGD with 0.9 momentum and 0.0005 weight decay. We set the initial learning rate to 10^{-3}, and use slightly different learning rate decay policy for different dataset, which will be described in details later.

优化。如上所述，RefineDet 的主干网络（如：VGG-16 和 ResNet-101）已经预先在 ILSVRC CLS-LOC 数据集 [37]上训练。我们使用"xavier" 方法 [17]来随机初始化基于 VGG-16 的 RefineDet 的两个额外增加的卷积层（即，conv6 1 和 conv6 2）中的参数，并为额外的 ResNet-101 残差块（即，res6）从均值为 0，标准差为 0.01 的高斯分布中提取参数。我们设置训练中的默认的批次大小为 32，然后使用 SGD 进行微调，momentum 为 0.9，权重衰减率为 0.0005.我们设置初始的学习率（the learning rate）为 10-3，并针对不同的数据集使用稍微不同的学习率衰减策略，稍后将对其进行详细描述。

Inference. At inference phase, the ARM first filters out the regularly tiled anchors with the negative confidence scores larger than the threshold θ, and then refines the locations and sizes of remaining anchors. After that, the ODM takes over these refined anchors, and outputs top 400 high confident detections per image. Finally, we apply the nonmaximum suppression with jaccard overlap of 0.45 per class and retain the top 200 high confident detections per image to produce the final detection results.

推理。在推理阶段，ARM 首先过滤掉那些具有高于某一阈值θ的负样本置信度的规则平铺的 anchors，然后对剩下的 anchors 的位置和大小进行细化。之后，ODM 获取这些经过细化的 anchors，并且对每张图像输出最高的 400 个置信度的结果。最后，我们应用具有 0.45 的 jaccard 重叠率的非最大抑制（the non-maximum suppression）并为每张图像保留 200 个具有最高置信度检测的结果，以产生最终检测结果。

5. Experiments

Experiments are conducted on three datasets: PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO. The PASCAL VOC and MS COCO datasets include 20 and 80 object classes, respectively. The classes in PASCAL VOC are the subset of that in MS COCO. We implement RefineDet in Caffe [23]. All the training and testing codes and the trained models are available at ：
https://github.com/ sfzhang15/RefineDet.

5、实验

实验在三个数据集上进行：PASCAL VOC 2007，PASCAL VOC 2012 和 MS COCO。 PASCAL VOC 和 MS COCO 数据集分别包括 20 个和 80 个对象类。PASCAL VOC 中的类别时 MS COCO 的一个子集。我们利用 Caffe [23]来实现 RefineDet。所有的训练和测试代码以及训练好的模型可以在以下网址获得：
https://github.com/ sfzhang15/RefineDet.

5.1 PASCAL VOC 2007

All models are trained on the VOC 2007 and VOC 2012 trainval sets, and tested on the VOC 2007

5.1 PASCAL VOC 2007

所有的模型都在 VOC 2007 和 VOC 2012 训练验证集上训练，在 VOC 2007 测试集上进行测试，

test set. We set the learning rate to 10^{-3} for the first $80k$ iterations, and decay it to 10^{-4} and 10^{-5} for training another $20k$ and $20k$ iterations, respectively. We use the default batch size 32 in training, and only use VGG-16 as the backbone network for all the experiments on the PASCAL VOC dataset, including VOC 2007 and VOC 2012.

We compare RefineDet[8] with the state-of-the-art detectors in Table 1. With low dimension input (*i.e.*, 320×320), RefineDet produces 80.0% mAP without bells and whistles, which is the first method achieving above 80% mAP with such small input images, much better than several modern objectors. By using larger input size 512×512, RefineDet achieves 81.8% mAP, surpassing all one-stage methods, *e.g.*, RON384 [24], SSD513 [13], DSSD513 [13], etc. Comparing to the two-stage methods, RefineDet512 performs better than most of them except CoupleNet [54], which is based on ResNet-101 and uses larger input size (*i.e.*, $\sim 1000 \times 600$) than our RefineDet512. As pointed out in [21], the input size significantly influences detection accuracy. The reason is that high resolution inputs make the detectors "seeing" small objects clearly to increase successful detections. To reduce the impact of input size for a fair comparison, we use the multi-scale testing strategy to evaluate RefineDet, achieving 83.1% (RefineDet320+) and 83.8% (RefineDet512+) mAPs, which are much better than the state-of-the-art methods.

5.1.1 Run Time Performance

We present the inference speed of RefineDet and the state-of-the-art methods in the fifth column of Table 1. The speed is evaluated with batch size 1 on a machine with NVIDIA Titan X, CUDA 8.0 and cuDNN v6. As shown in Table 1, we find that RefineDet processes an image in 24.8ms (40.3 FPS) and 41.5ms (24.1 FPS) with input sizes 320×320 and 512×512, respectively. To the best of our knowledge, RefineDet is the first real-time

我们将前 80k 次迭代的学习率为设置为 10-3，接下来的 20k 次迭代，学习率衰减至 10-4，再接下来的 20k 次迭代，学习率衰减至 10-5。我们使用默认大小为 32 的批次（batch）进行训练，在所有的 PASCAL VOC 数据集上进行的实验，包括 VOC 2007 和 VOC 2012，都使用 VGG-16 作为主干网络。

我们把 RefineDet 和表 1 中最先进的检测器进行了比较。在低维输入的情况下（即， 320 x 320），RefineDet 在没有使用其他措施的条件下达到了 80.0% mAP，这也是第一款检测器可以在如此小的输入图像的情况下达到 80.0% mAP，远优于几个先进的目标检测器。使用更大的输入尺寸 512 x 512，RefineDet 达到了 81.8% mAP，超过了所有的采用单阶段设计的方法，如：RON384 [24], SSD513 [13], DSSD513 [13]等。和采用两阶段设计的方法相比，RefineDet512 的表现比大多数都要好，除了 CoupleNet [54]，CoupleNet 基于 ResNet-101 设计，并使用比 RefineDet 更大的输入尺寸（即，~1000 x 600）。正如参考文献[21]中指出，输入大小会在很大程度上影响检测的准确性。原因是高分辨率的输入会让检测器清楚地"看见"小目标，从而增加检测成功率。为了减少输入尺寸带来的影响，以保证比较的公平性，我们使用多尺度测试策略来评估 RefineDet，最后达到了 83.1% mAP (RefineDet320+) 和 83.8% (RefineDet512+) mAP，这比现有的最先进的方法要好。

5.1.1 运行时间性能

我们在表 1 的第五列中提供了 RefineDet 和最先进的方法的推理速度。该速度为在具有 NVIDIA Titan X，CUDA 8.0 和 cuDNN v6 的机器上，批量大小为 1 时评估速度。如表 1 中所示，我们发现，在输入大小分别为 320 x 320 和 512 X 512 的情况下，RefineDet 处理一张图片的时间分别是 24.8ms (40.3 FPS) 和 41.5ms (24.1 FPS)。据我们所知，RefineDet 是第一个检测准确率达到 80% mAP 的实时监测方法。与 SSD, RON，DSSD，DSOD 相比，

[8] Due to the shortage of computational resources, we only train RefineDet with two kinds of input size, *i.e.*, 320×320 and 512×512. We believe the accuracy of RefineDet can be further improved using larger input images.

method to achieve detection accuracy above 80% mAP on PASCAL VOC 2007. Comparing to SSD, RON, DSSD and DSOD, RefineDet associates fewer anchor boxes on the feature maps (*e.g.*, 24564 anchor boxes in SSD512*[30] *vs.* 16320 anchor boxes in RefineDet512). However, RefineDet still achieves top accuracy with high efficiency, mainly thanks to the design of two inter-connected modules, (*e.g.*, two-step regression), which enables RefineDet to adapt to different scales and aspect ratios of objects. Meanwhile, only YOLO and SSD300* are slightly faster than our RefineDet320, but their accuracy are 16.6% and 2.5% worse than ours. In summary, RefineDet achieves the best trade-off between accuracy and speed.

RefineDet 可以在特征图上关联更少的 anchor boxes（例如，SSD512*[30]中有 24564 个 anchor boxes，RefineDet512 有 16320 个 anchor boxes）。然而，RefineDet 仍然以高效率实现了高精度，这主要得益于两个互联模块的设计（如，两步回归），这使得 RefineDet 能够使用目标的不同比例和纵横比。同时，只有 YOLO 和 SSD300 *比我们的 RefineDet320 略快一些，但其精度比我们的低 16.6%和 2.5%。 总之，RefineDet 在精度和速度之间实现了最佳平衡。

Table 1: Detection results on PASCAL VOC dataset. For VOC 2007, all methods are trained on VOC 2007 and VOC 2012 trainval sets and tested on VOC 2007 test set. For VOC 2012, all methods are trained on VOC 2007 and VOC 2012 trainval sets plus VOC 2007 test set, and tested on VOC 2012 test set. Bold fonts indicate the best mAP.

表 1：PASCAL VOC 数据集上的检测结果。对于 VOC 2007，所有的方法都在 VOC 2007 和 VOC 2012 训练验证集上进行训练，并在 VOC 2007 测试集上进行测试。对于 VOC 2012，所有的方法都在 VOC 2007 和 VOC 2012 训练验证集外加 VOC 2007 测试集上进行训练，并在 VOC 2012 测试集上进行测试。表格中粗体字表示最佳 mAP。

Method	Backbone	Input size	#Boxes	FPS	mAP (%)	
					VOC 2007	VOC 2012
two-stage:						
Fast R-CNN[15]	VGG-16	$\sim 1000 \times 600$	~ 2000	0.5	70.0	68.4
Faster R-CNN[36]	VGG-16	$\sim 1000 \times 600$	300	7	73.2	70.4
OHEM[41]	VGG-16	$\sim 1000 \times 600$	300	7	74.6	71.9
HyperNet[25]	VGG-16	$\sim 1000 \times 600$	100	0.88	76.3	71.4
Faster R-CNN[36]	ResNet-101	$\sim 1000 \times 600$	300	2.4	76.4	73.8
ION[1]	VGG-16	$\sim 1000 \times 600$	4000	1.25	76.5	76.4
MR-CNN[14]	VGG-16	$\sim 1000 \times 600$	250	0.03	78.2	73.9
R-FCN[5]	ResNet-101	$\sim 1000 \times 600$	300	9	80.5	77.6
CoupleNet[54]	ResNet-101	$\sim 1000 \times 600$	300	8.2	82.7	80.4
one-stage:						
YOLO[34]	GoogleNet [45]	448×448	98	45	63.4	57.9
RON384[24]	VGG-16	384×384	30600	15	75.4	73.0
SSD321[13]	ResNet-101	321×321	17080	11.2	77.1	75.4
SSD300*[30]	VGG-16	300×300	8732	46	77.2	75.8
DSOD300[39]	DS/64-192-48-1	300×300	8732	17.4	77.7	76.3
YOLOv2[35]	Darknet-19	544×544	845	40	78.6	73.4
DSSD321[13]	ResNet-101	321×321	17080	9.5	78.6	76.3
SSD512*[30]	VGG-16	512×512	24564	19	79.8	78.5
SSD513[13]	ResNet-101	513×513	43688	6.8	80.6	79.4
DSSD513[13]	ResNet-101	513×513	43688	5.5	81.5	80.0
RefineDet320	VGG-16	320×320	6375	40.3	80.0	78.1
RefineDet512	VGG-16	512×512	16320	24.1	81.8	80.1
RefineDet320+	VGG-16	-	-	-	83.1	82.7
RefineDet512+	VGG-16	-	-	-	**83.8**	**83.5**

5.1.2 Ablation Study

5.1.2 消融研究

To demonstrate the effectiveness of different components in RefineDet, we construct four variants and evaluate them on VOC 2007, shown in Table 3. Specifically, for a fair comparison, we use the same parameter settings and input size (320×320) in evaluation. All models are trained on VOC 2007 and VOC 2012 trainval sets, and tested on VOC 2007 test set.

Negative Anchor Filtering. To demonstrate the effectiveness of the negative anchor filtering, we set the confidence threshold θ of the anchors to be negative to 1.0 in both training and testing. In this case, all refined anchors will be sent to the ODM for detection. Other parts of RefineDet remain unchanged. Removing negative anchor filtering leads to 0.5% drop in mAP (*i.e.*, 80.0% *vs.* 79.5%). The reason is that most of these well-classified negative anchors will be filtered out during training, which solves the class imbalance issue to some extent.

Two-Step Cascaded Regression. To validate the effectiveness of the two-step cascaded regression, we redesign the network structure by directly using the regularly paved anchors instead of the refined ones from the ARM (see the fourth column in Table 3). As shown in Table 3, we find that mAP is reduced from 79.5% to 77.3%. This sharp decline (*i.e.*, 2.2%) demonstrates that the two-step anchor cascaded regression significantly help promote the performance.

Transfer Connection Block. We construct a network by cutting the TCBs in RefineDet and redefining the loss function in the ARM to directly detect multi-class of objects, just like SSD, to demonstrate the effect of the TCB. The detection accuracy of the model is presented in the fifth column in Table 3. We compare the results in the fourth and fifth columns in Table 3 (77.3% *vs.* 76.2%) and find that the TCB improves the mAP by 1.1%. The main reason is that the model can inherit the discriminative features from the ARM, and integrate large-scale context information to improve the detection accuracy by using the TCB.

为了表明 RefineDet 中不同组建的有效性，我们构建了四个变体，并在 VOC 2007 数据集上对它们进行评估。为了比较的公平性，我们在评估时使用相同的参数设置和输入大小（320 x 320）。所有的模型都在 VOC 2007 和 VOC 2012 训练验证集上进行训练，在 VOC 2007 测试集上进行测试。

负样本 anchor 过滤机制。为了证明负样本 anchor 过滤机制的有效性，我们在训练和测试过程中将 anchors 的置信度阈值θ都设置为负值 1.0。在这种情况下，所有的细化后的 anchors 都会被送入 ODM 进行检测。RefineDet 的其他部分保持不变。去除负样本 anchor 过滤机制后，mAP 下降了下降 0.5％（即，80.0％比 79.5％）。原因在于大多数被分类的负样本 anchors 都会在训练过程中被过滤掉，从而在一定程度上解决类别失衡问题。

两步级联回归。为了验证两步级联回归的有效性，我们重新设计了网络结构，直接使用规则平铺的 anchors 而不是来自 ARM 经过细化的 anchors（见表 3 第四列）。如表 3 所示，我们发现 mAP 从 79.5％降至 77.3％。 这种急剧下降（即 2.2％）表明，两步锚定级联回归显着有助于提升性能。

传输连接块。我们通过移除 RefineDet 中的 TCB 并重新定义 ARM 中的损失函数来构建网络，以直接检测多类目标来，同 SSD 一样，展示 TCB 的效果。模型的检测精度见表 3 第五列。我们比较表 3 中第四和第五列的结果（77.3％和 76.2％），发现 TCB 将 mAP 提高了 1.1％。主要原因是该模型能够继承 ARM 的判别特征，并结合大规模的上下文信息来提高 TCB 的检测精度。

Table 2: Detection results on MS COCO test-dev set. Bold fonts indicate the best performance.

表 1： MS COCO test-dev 数据集上的检测结果。粗体字表示最佳性能。

Method	Data	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage:								
Fast R-CNN [15]	train	VGG-16	19.7	35.9	-	-	-	-
Faster R-CNN [36]	trainval	VGG-16	21.9	42.7	-	-	-	-
OHEM [41]	trainval	VGG-16	22.6	42.5	22.2	5.0	23.7	37.9
ION [1]	train	VGG-16	23.6	43.2	23.6	6.4	24.1	38.3
OHEM++ [41]	trainval	VGG-16	25.5	45.9	26.1	7.4	27.7	40.3
R-FCN [5]	trainval	ResNet-101	29.9	51.9	-	10.8	32.8	45.0
CoupleNet [54]	trainval	ResNet-101	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN by G-RMI [21]	-	Inception-ResNet-v2[44]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN+++ [19]	trainval	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	trainval35k	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w TDM [42]	trainval	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
Deformable R-FCN [6]	trainval	Aligned-Inception-ResNet	37.5	58.0	40.8	19.4	40.1	52.5
umd_det [2]	trainval	ResNet-101	40.8	62.4	44.9	23.0	43.4	53.2
G-RMI [21]	trainval32k	Ensemble of Five Models	41.6	61.9	45.4	23.9	43.5	54.9
one-stage:								
YOLOv2 [35]	trainval35k	DarkNet-19[35]	21.6	44.0	19.2	5.0	22.4	35.5
SSD300* [30]	trainval35k	VGG-16	25.1	43.1	25.8	6.6	25.9	41.4
RON384++ [24]	trainval	VGG-16	27.4	49.5	27.1	-	-	-
SSD321 [13]	trainval35k	ResNet-101	28.0	45.4	29.3	6.2	28.3	49.3
DSSD321 [13]	trainval35k	ResNet-101	28.0	46.1	29.2	7.4	28.1	47.6
SSD512* [30]	trainval35k	VGG-16	28.8	48.5	30.3	10.9	31.8	43.5
SSD513 [13]	trainval35k	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [13]	trainval35k	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet500 [28]	trainval35k	ResNet-101	34.4	53.1	36.8	14.7	38.5	49.1
RetinaNet800 [28]*	trainval35k	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RefineDet320	trainval35k	VGG-16	29.4	49.2	31.3	10.0	32.0	44.4
RefineDet512	trainval35k	VGG-16	33.0	54.5	35.5	16.3	36.3	44.3
RefineDet320	trainval35k	ResNet-101	32.0	51.4	34.2	10.5	34.7	50.4
RefineDet512	trainval35k	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
RefineDet320+	trainval35k	VGG-16	35.2	56.1	37.7	19.5	37.2	47.0
RefineDet512+	trainval35k	VGG-16	37.6	58.7	40.8	22.7	40.3	48.3
RefineDet320+	trainval35k	ResNet-101	38.6	59.9	41.7	21.1	41.7	52.3
RefineDet512+	trainval35k	ResNet-101	**41.8**	**62.9**	**45.7**	**25.6**	**45.1**	**54.1**

Table 3: Effectiveness of various designs. All models are trained on VOC 2007 and VOC 2012 trainval set and tested on VOC 2007 test set.

表 3：各种网络设计的有效性。所有的模型都在 VOC 2007 和 VOC 2012 训练验证集上进行训练，再 VOC 2007 测试集上进行测试。

Component	RefineDet320			
negative anchor filtering?	✓			
two-step cascaded regression?	✓	✓		
transfer connection block?	✓	✓	✓	
mAP (%)	80.0	79.5	77.3	76.2

5.2 PASCAL VOC 2012

Following the protocol of VOC 2012, we submit the detection results of RefineDet to the public testing server for evaluation. We use VOC 2007 trainval set and test set plus VOC 2012 trainval set (21,503 images) for training, and test on VOC 2012 test set (10,991 images). We use the default batch size 32 in training. Meanwhile, we set the learning rate to 10^{-3} in the first $160k$ iterations, and decay it to 10^{-4} and 10^{-5} for another $40k$ and $40k$ iterations.

5.2 PASCAL VOC 2012

遵循 VOC 2012 协议，我们将 RefineDet 的检测结果提交给公共测试服务器进行评估。我们使用 VOC 2007 训练集和测试集以及 VOC 2012 训练集（21,503 幅图像）进行训练，并在 VOC 2012 测试集（10,991 幅图像）上进行测试。同时，我们在第一个 160k 次迭代中将学习速率设置为 10-3，并且将其衰减到 10-4 以进行第一个 40k 次迭代，再将其衰减到 10-5 以进行另外的 40k 迭代。

Table 1 shows the accuracy of the proposed RefineDet algorithm, as well as the state-of-the-art methods. Among the methods fed with input size 320×320, RefineDet320 obtains the top 78.1% mAP, which is even better than most of those two-stage methods using about 1000×600 input size (*e.g.*, 70.4% mAP of Faster R-CNN [36] and 77.6% mAP of R-FCN [5]). Using the input size 512×512, RefineDet improves mAP to 80.1%, which is surpassing all one-stage methods and only slightly lower than CoupleNet [54] (*i.e.*, 80.4%). CoupleNet uses ResNet-101 as base network with 1000×600 input size. To reduce the impact of input size for a fair comparison, we also use multi-scale testing to evaluate RefineDet and obtain the state-of-the-art mAPs of 82.7% (RefineDet320+) and 83.5% (RefineDet512+).

5.3 MS COCO

In addition to PASCAL VOC, we also evaluate RefineDet on MS COCO [29]. Unlike PASCAL VOC, the detection methods using ResNet-101 always achieve better performance than those using VGG-16 on MS COCO. Thus, we also report the results of ResNet-101 based RefineDet. Following the protocol in MS COCO, we use the trainval35k set [1] for training and evaluate the results from test-dev evaluation server. We set the batch size to 32 in training[9], and train the model with 10^{-3} learning rate for the first $280k$ iterations, then 10^{-4} and 10^{-5} for another $80k$ and $40k$ iterations, respectively.

Table 7 shows the results on MS COCO test-dev set. RefineDet320 with VGG-16 produces 29.4% AP that is better than all other methods based on VGG-16 (*e.g.*, SSD512* [30] and OHEM++ [41]). The accuracy of RefineDet can be improved to 33.0% by using larger input size (*i.e.*, 512×512), which is much better than several modern object detectors, *e.g.*, Faster R-CNN [36] and SSD512* [30].

表 1 显示了建议的 RefineDet 算法以及最先进的方法的精确度。在输入大小为 320×320 的方法中，RefineDet320 获得最高 78.1％的 mAP，这比使用大约 1000×600 输入大小的采用两阶段设计的方法（例如，Faster R-CNN 的 70.4％ mAP [36]和 77.6％的 R-FCN 的 mAP [5]）的结果还要好。使用输入大小 512×512，RefineDet 将 mAP 提高到 80.1％，这超过了所有的单阶段设计的方法，并且仅略低于 CoupleNet[54]（即 80.4％）。CoupleNet 的输入尺寸为 1000 x 600，并使用 ResNet-101 作为基础网络。为了减少输入大小对比较公平性的影响，我们还使用多尺度测试来评估 RefineDet，并获得最新的 82.7％（RefineDet320 +）和 83.5％（RefineDet512 +）的 mAP。

5.3 MS COCO

除了 PASCAL VOC，我们还在 MS COCO[29]上评估 RefineDet。与 PASCAL VOC 不同，在 MS COCO 上，使用 ResNet-101 的检测方法总是比使用 VGG-16 的更好。因此，我们也报告了基于 ResNet-101 的 RefineDet 的结果。遵循 MS COCO 中的协议，我们使用 trainval35k 数据集[1]进行训练并评估来自 test-dev 评估服务器的结果。我们在训练中将批处理大小设置为 32，并且在前 280k 迭代中以 10-3 学习率训练模型，然后分别以 10-4 和 10-5 进行另一个 80k 和 40k 迭代。

表 7 展示了 MS COCO test-dev 集合的结果。使用 VGG-16 的 RefineDet320 产生的 29.4％的 AP，这个结果比基于 VGG-16 的所有其他方法（例如 SSD512 * [30]和 OHEM ++ [41]）要好。通过使用更大的输入尺寸（即 512 x 512），RefineDet 的精确度可以提高的 33.0%，这个结果比比很多目标检测器要好，例如：Faster R-CNN [36] 和 SSD512* [30]。同时，使用 ResNet-101 可以更一步提高 RefineDet

[9] Due to the memory issue, we reduce the batch size to 20 (which is the largest batch size we can use for training on a machine with 4 NVIDIA M40 GPUs) to train the ResNet-101 based RefineDet with the input size 512×512, and train the model with 10^{-3} learning rate for the first $400k$ iterations, then 10^{-4} and 10^{-5} for another $80k$ and $60k$ iterations.

Meanwhile, using ResNet-101 can further improve the performance of RefineDet, *i.e.*, RefineDet320 with ResNet-101 achieves 32.0% AP and RefineDet512 achieves 36.4% AP, exceeding most of the detection methods except Faster RCNN w TDM [42], Deformable R-FCN [6], RetinaNet800 [28], umd det [2], and G-RMI [21]. All these methods use a much bigger input images for both training and testing (*i.e.*, 1000×600 or 800×800) than our RefineDet (*i.e.*, 320×320 and 512 × 512). Similar to PASCAL VOC, we also report the multi-scale testing AP results of RefineDet for fair comparison in Table 7, *i.e.*, 35.2% (RefineDet320+ with VGG16), 37.6% (RefineDet512+ with VGG-16), 38.6% (RefineDet320+ with ResNet-101) and 41.8% (RefineDet512+ with ResNet-101). The best performance of RefineDet is 41.8%, which is the state-of-the-art, surpassing all published two-stage and one-stage approaches. Although the second best detector G-RMI [21] ensembles five Faster RCNN models, it still produces 0.2% lower AP than RefineDet using a single model. Comparing to the third and fourth best detectors, *i.e.*, umd det [2] and RetinaNet800 [28], RefineDet produces 1.0% and 2.7% higher APs. In addition, the main contribution: focal loss in RetinaNet800, is complementary to our method. We believe that it can be used in RefineNet to further improve the performance.

5.4 From MS COCO to PASCAL VOC

We study how the MS COCO dataset help the detection accuracy on PASCAL VOC. Since the object classes in PASCAL VOC are the subset of MS COCO, we directly fine-tune the detection models pretrained on MS COCO via subsampling the parameters, which achieves 84.0% mAP (RefineDet320) and 85.2% mAP (RefineDet512) on VOC 2007 test set, and 82.7% mAP (RefineDet320) and 85.0% mAP (RefineDet512) on VOC 2012 test set, shown in Table 4. After using the multi-scale testing, the detection accuracy are promoted to 85.6%, 85.8%, 86.0% and 86.8%, respectively. As

的性能，即使用 ResNet-101 的 RefinDet320 可以达到 32.0%的 AP，RefineDet512 达到了 36.4%的 AP，这个结果已经超过了大多数检测方法，除了 Faster RCNN w TDM [42]， Deformable R-FCN [6]，RetinaNet800 [28]，umd det [2]，and G-RMI [21]。所有的这些方法使用比 RefineDet 更大的训练和测试输入图像尺寸（即，1000×600 或 800×800），Refine 使用 320×320 和 512×512 大小的图像输入尺寸。和 PASCAL VOC 类似，我们也报告了 RefineDet 在表 7 中公平比较的多尺度测试 AP 结果，即 35.2%（RefineDet320 +使用 VGG-16 模型），37.6%（RefineDet512 +使用 VGG-16 模型），38.6%（RefineDet320 +使用 ResNet -101 模型）和 41.8%（使用 ResNet-101 的 RefineDet512 +）。RefineDet 最高的结果为 41.8%，这已经是当前最佳的结果，超过了已经发表的所有的采用但阶段设计和两阶段设计的方法。虽然排名第二好的检测器 G-RMI [21]集成了 5 个 Faster RCNN 模型，但它仍然比使用单一模型的 RefineDet 减少 0.2%的 AP。与第三和第四个最佳检测器，即，RefineMet det [2]和 RetinaNet800 [28]相比，RefineDet 产生的 AP 更高 1.0%和 2.7%。 另外，主要贡献：RetinaNet800 的焦点损失，与我们的方法相辅相成。 我们相信它可以在 RefineNet 中使用以进一步提高性能。

5.4 从 MS COCO 到 PASCAL VOC

我们研究 MS COCO 数据集如何帮助 PASCAL VOC 的检测精度。由于 PASCAL VOC 中的对象类别是 MS COCO 的子集，因此我们通过对参数进行二次抽样，直接对 MS COCO 预先训练的检测模型进行微调，在 VOC 2007 测试集上达到 84.0%mAP（RefineDet320）和 85.2%mAP（RefineDet512），在 VOC 2012 测试集上达到了 82.7%的 mAP（RefineDet320）和 85.0%的 mAP（RefineDet512），结果如表 4 所示。使用多尺度检测后，检测准确度分别提高到 85.6%，85.8%，86.0%和 86.8%。 如表 4 所示，使用 MS COCO 和 PASCAL VOC 中的训练数据后，我们的 RefineDet 获得了 VOC 2007 和 VOC

shown in Table 4, using the training data in MS COCO and PASCAL VOC, our RefineDet obtains the top mAP scores on both VOC 2007 and VOC 2012. Most important, our single model RefineNet512+ based on VGG16 ranks as the top 5 on the VOC 2012 Leaderboard (see [9]), which is the best accuracy among all one-stage methods. Other two-stage methods achieving better results are based on much deeper networks (*e.g.*, ResNet-101 [19] and ResNeXt-101 [49]) or using ensemble mechanism.

2012 上的最高 mAP 分数。最重要的是，我们基于 VGG16 的单一模型 RefineNet512 +在 VOC 2012 排行榜上排名前五（见[9]），这是所有采用单一阶段设计的方法中最准确的。其他的采用两阶段设计的模型取得了更好的结果，因为它们是采用了更深的网络结构（例如，ResNet-101 [19]和 ResNeXt-101 [49]），或者使用了集成机制。

Table 4: Detection results on PASCAL VOC dataset. All models are pre-trained on MS COCO, and fine-tuned on PASCAL VOC. Bold fonts indicate the best mAP.

表 4：PASCAL VOC 数据集检测结果。所有的模型都预先在 MS COCO 数据集上进行训练，然后再 PASCAL VOC 数据集上进行微调。粗体字表明最佳 mAP。

Method	Backbone	mAP (%)	
		VOC 2007 test	VOC 2012 test
two-stage:			
Faster R-CNN[36]	VGG-16	78.8	75.9
OHEM++[41]	VGG-16	-	80.1
R-FCN[5]	ResNet-101	83.6	82.0
one-stage:			
SSD300[30]	VGG-16	81.2	79.3
SSD512[30]	VGG-16	83.2	82.2
RON384++[24]	VGG-16	81.3	80.7
DSOD300[39]	DS/64-192-48-1	81.7	79.3
RefineDet320	VGG-16	84.0	82.7
RefineDet512	VGG-16	85.2	85.0
RefineDet320+	VGG-16	85.6	86.0
RefineDet512+	VGG-16	**85.8**	**86.8**

6 Conclusion

In this paper, we present a single-shot refinement neural network based detector, which consists of two interconnected modules, *i.e.*, the ARM and the ODM. The ARM aims to filter out the negative anchors to reduce search space for the classifier and also coarsely adjust the locations and sizes of anchors to provide better initialization for the subsequent regressor, while the ODM takes the refined anchors as the input from the former ARM to regress the accurate object locations and sizes

6 结论

在本文中，我们提出了一个基于单发（single-shot）细化神经网络的检测器，它由两个互连的模块组成，即 ARM 和 ODM。ARM 旨在过滤出负样本 anchors 以减少分类器的搜索空间，并粗略地调整 anchors 的位置和大小，以便为后续的回归器提供更好的初始化。而 ODM 则将来自 ARM 的细化后的 anchors 作为输入，回归准确的目标位置和尺寸，并预测相应的多类标签。整个网络使用多任务损失，并采用端到端的方式进行训练。我们在 PASCAL VOC 2007，PASCAL VOC 2012 和 MS COCO 数据集上

and predict the corresponding multiclass labels. The whole network is trained in an end-to-end fashion with the multi-task loss. We carry out several experiments on PASCAL VOC 2007, PASCAL VOC 2012, and MS COCO datasets to demonstrate that RefineDet achieves the state-of-the-art detection accuracy with high efficiency. In the future, we plan to employ RefineDet to detect some other specific kinds of objects, *e.g.*, pedestrian, vehicle, and face, and introduce the attention mechanism in RefineDet to further improve the performance.

进行了多项实验，以证明 RefineDet 能够高效地实现目前最好的检测精度。未来，我们计划使用 RefineDet 来检测其他特定类型的物体，例如行人，车辆和人脸，并在 RefineDet 中引入注意力（attention）机制以进一步提高性能。

Reference 参考文献

[1] S. Bell, C. L. Zitnick, K. Bala, and R. B. Girshick. Insideoutside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, pages 2874–2883, 2016. 3, 6, 7, 8

[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Improving object detection with one line of code. In *ICCV*, 2017. 7, 8

[3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, pages 354–370, 2016. 1, 3

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 4

[5] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 1, 3, 6, 7, 8

[6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017. 7, 8

[7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2155–2162, 2014. 4

[8] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 1, 3

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Leaderboard of the PASCAL Visual Object Classes Challenge 2012 (VOC2012). http://host.robots.ox.ac.uk:8080/leaderboard/ displaylb.php?challengeid=11&compid=4. Online; accessed 1 October 2017. 8

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/ voc2007/workshop/index.html. Online; accessed 1 October 2017. 2

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http:

//www.pascal-network.org/challenges/VOC/ voc2012/workshop/index.html. Online; accessed 1 October 2017. 2, 3

[12] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010. 3

[13] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017. 3, 5, 6, 7

[14] S. Gidaris and N. Komodakis. Object detection via a multiregion and semantic segmentation-aware CNN model. In *ICCV*, pages 1134–1142, 2015. 3, 6

[15] R. B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015. 1, 3, 5, 6, 7

[16] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 3

[17] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010. 5

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361, 2014. 3

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 4, 7, 8

[20] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *CoRR*, abs/1312.5402, 2013. 4

[21] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara,
A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017. 5, 7, 8

[22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 4

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, pages 675–678, 2014. 5

[24] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen. RON: reverse connection with objectness prior networks for object detection. In *CVPR*, 2017. 1, 3, 5, 6, 7, 8

[25] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *CVPR*, pages 845–853, 2016. 3, 6

[26] H. Lee, S. Eum, and H. Kwon. ME R-CNN: multi-expert region-based CNN for object detection. In *ICCV*, 2017. 3

[27] T. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and´ S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 3, 7

[28] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollar. Focal´ loss for dense object detection. In *ICCV*, 2017. 1, 3, 7, 8

[29] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: com-´ mon objects in context. In *ECCV*, pages 740–755, 2014. 1, 2, 3, 8

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1, 3, 4, 6, 7, 8

[31] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. In *ICLR workshop*, 2016. 4

[32] P. H. O. Pinheiro, R. Collobert, and P. Dollar. Learning to´ segment object candidates. In *NIPS*, pages 1990–1998, 2015. 3

[33] P. O. Pinheiro, T. Lin, R. Collobert, and P. Dollar. Learning´ to refine object segments. In *ECCV*, pages 75–91, 2016. 3

[34] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 3, 6

[35] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. 1, 3, 6, 7

[36] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2017. 1, 3, 6, 7, 8

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh,
S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 3, 4, 5

[38] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 3

[39] Z. Shen, Z. Liu, J. Li, Y. Jiang, Y. Chen, and X. Xue. DSOD: learning deeply supervised object detectors from scratch. In *ICCV*, 2017. 3, 6, 8

[40] A. Shrivastava and A. Gupta. Contextual priming and feedback for faster R-CNN. In *ECCV*, pages 330–348, 2016. 3

[41] A. Shrivastava, A. Gupta, and R. B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016. 1, 3, 6, 7, 8

[42] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *CoRR*, abs/1612.06851, 2016. 3, 7, 8

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 4

[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 4, 7

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 6

[46] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 3

[47] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511– 518, 2001. 3

[48] X. Wang, A. Shrivastava, and A. Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *CVPR*, 2017. 3

[49] S. Xie, R. B. Girshick, P. Dollar, Z. Tu, and K. He. Aggre-´ gated residual transformations for deep neural networks. In *CVPR*, 2017. 4, 8

[50] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang. Gated bi-directional CNN for object detection. In *ECCV*, pages 354–369, 2016. 3

[51] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. Detecting face with densely connected face proposal network. In *CCBR*, pages 3–12, 2017. 4

[52] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. Faceboxes: A CPU real-time face detector with high accuracy. In *IJCB*, 2017. 4

[53] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S^3FD: Single shot scale-invariant face detector. In *ICCV*, 2017. 1, 3, 4

[54] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. In *ICCV*, 2017. 3, 5, 6, 7

[55] C. L. Zitnick and P. Dollar.´ Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014. 3

7. Complete Object Detection Results

We show the complete object detection results of the proposed RefineDet method on the PASCAL VOC 2007 test set, PASCAL VOC 2012 test set and MS COCO test-dev set in Table 5, Table 6 and Table 7, respectively. Among the results of all published methods, our RefineDet achieves the best performance on these three detection datasets, *i.e.*, 85.8% mAP on the PASCAL VOC 2007 test set, 86.8% mAP on the PASCAL VOC 2012 test set and 41.8% AP on the MS COCO test-dev set.

8. Qualitative Results

We show some qualitative results on the PASCAL VOC 2007 test set, the PASCAL VOC 2012 test set and the MS COCO test-dev in Figure 3, Figure 4, and Figure 5, respectively. We only display the detected bounding boxes with the score larger than 0.6. Different colors of the bounding boxes indicate different object categories. Our method works well with the occlusions, truncations, inter-class interference and clustered background.

9. Detection Analysis on PASCAL VOC 2007

We use the detection analysis tool [10] to understand the performance of two RefineDet

[10] http://web.engr.illinois.edu/~dhoiem/projects/ detectionAnalysis/

7. 完整的目标检测实验结果

表格 5, 6, 7 分别显示了我们提出的 RefineDet 方法在 PASCAL VOC 2007 测试集, PASCAL VOC 2012 测试集和 MS COCO test-dev 测试集上的结果。在所有已经被公布的方法的结果中，我们的 RefineDet 在三个目标检测数据集上都达到了目前最好的结果，即在 PASCAL VOC 2007 数据集上达到了 85.8% mAP，在 PASCAL VOC 2012 数据集上达到了 86.8% mAP，在 MS COCO test-dev 测试机上达到了 41.8% AP。

8. 定性结果

我们分别在图 3，图 4 和图 5 中显示了在 PASCAL VOC 2007 测试集，PASCAL VOC 2012 测试集和 MS COCO 测试集的一些定性结果。我们只显示分数大于 0.6 的检测到的边界框。边界框的不同颜色表示不同的对象类别。我们的方法适用于遮挡，截断，类间干扰和聚类背景。

9. PASCAL VOC 2007 检测分析

我们使用检测分析工具清楚地了解了两个 RefineDet 模型（即 RefineDet320 和 RefineDet512）

models (*i.e.*, RefineDet320 and RefineDet512) clearly. Figure 6 shows that RefineDet can detect various object categories with high quality (large white area). The majority of its confident detections are correct. The recall is around 95%-98%, and is much higher with "weak" (0.1 jaccard overlap) criteria. Compared to SSD, RefineDet reduces the false positive errors at all aspects: (1) RefineDet has less localization error (Loc), indicating that RefineDet can localize objects better because it uses two-step cascade to regress the objects. (2) RefineDet has less confusion with background (BG), due to the negative anchor filtering mechanism in the anchor refinement module (ARM). (3) RefineDet has less confusion with similar categories (Sim), benefiting from using two-stage features to describe the objects, *i.e.*, the features in the ARM focus on the binary classification (being an object or not), while the features in the object detection module (ODM) focus on the multi-class classification (background or object classes).

Figure 7 demonstrates that RefineDet is robust to different object sizes and aspect ratios. This is not surprising because the object bounding boxes are obtained by the twostep cascade regression, *i.e.*, the ARM diversifies the default scales and aspect ratios of anchor boxes so that the ODM is able to regress tougher objects (*e.g.*, extra-small, extralarge, extra-wide and extra-tall). However, as shown in Figure 7, there is still much room to improve the performance of RefineDet for small objects, especially for the chairs and tables. Increasing the input size (*e.g.*, from 320 × 320 to 512 × 512) can improve the performance for small objects , but it is only a temporary solution. Large input will be a burden on running speed in inference. Therefore, detecting small objects is still a challenge task and needs to be further studied.

的性能。图 6 显示了 RefineDet 可以检测出高质量（白色区域）的各种类别对象。大部分置信度较高的检测结果都是正确的。召回率在 95%-98%左右，而"弱"（0.1 jaccard 重叠）标准的召回率要高得多。与 SSD 相比，RefineDet 减少了所有方面的误报错误：（1）RefineDet 定位误差较小（Loc），表明 RefineDet 可以更好地定位对象，因为它使用两步级联来回归对象。（2）由于 anchor 细化模块（ARM）中的负样本 anchor 过滤机制，RefineDet 与背景（BG）的混淆较少。（3）RefineDet 与类似类（Sim）的混淆较少，受益于使用两阶段特征来描述对象，即 ARM 中的特征侧重于二元分类（是否为对象），而特征 对象检测模块（ODM）专注于多类分类（背景或对象类）。

图 7 演示了 RefineDet 对不同大小和宽高比的对象检测的健壮性。 这并不令人惊讶，因为对象边界框是通过两步级联回归获得的，即 ARM 多样化 anchor 的默认比例和高宽比，以便 ODM 能够回归更困难的对象（例如，超小型，超大型， 超宽和超高）。但是，如图 7 所示，仍然有很多空间可以改善 RefineDet 对于小目标检测的性能，特别是对于椅子和桌子。增加输入尺寸（例如从 320×320 到 512×512）可以改善小物体的性能，但这只是一个临时解决方案。 大量输入将成为推断中运行速度的负担。 因此，探测小物体仍然是一个挑战，需要进一步研究。

Table 5: Object detection results on the PASCAL VOC 2007 test set. All models use VGG-16 as the backbone network.
表 5：PASCAL VOC 2007 测试机上的目标检测结果。所有的模型使用 VGG-16 作为主干网络。

Method	Data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
RefineDet320	07+12	80.0	83.9	85.4	81.4	75.5	60.2	86.4	88.1	89.1	62.7	83.9	77.0	85.4	87.1	86.7	82.6	55.3	82.7	78.5	88.1	79.4
RefineDet512	07+12	81.8	88.7	87.0	83.2	76.5	68.0	88.5	88.7	89.2	66.5	87.9	75.0	86.8	89.2	87.8	84.7	56.2	83.2	78.7	88.1	82.3
RefineDet320+	07+12	83.1	89.5	87.9	84.9	79.7	70.0	87.5	89.1	89.8	69.8	87.1	76.4	86.6	88.6	88.4	85.3	62.4	83.7	82.3	89.0	83.1
RefineDet512+	07+12	83.8	88.5	89.1	85.5	79.8	72.4	89.5	89.5	89.9	69.9	88.9	75.9	87.4	89.6	89.0	86.2	63.9	86.2	81.0	88.6	84.4
RefineDet320	COCO+07+12	84.0	88.9	88.4	86.2	81.5	71.7	88.4	89.4	89.0	71.0	87.0	80.1	88.5	90.2	88.4	86.7	61.2	85.2	83.8	89.1	85.5
RefineDet512	COCO+07+12	85.2	90.0	89.2	87.9	83.1	78.5	90.0	89.9	89.7	74.7	89.8	79.5	88.7	89.9	89.2	87.8	63.1	86.4	82.3	89.5	84.7
RefineDet320+	COCO+07+12	85.6	90.2	89.0	87.6	84.6	78.0	89.4	89.7	89.9	74.7	89.8	80.5	89.0	89.7	89.6	87.8	65.5	87.9	84.2	88.6	86.3
RefineDet512+	COCO+07+12	85.8	90.4	89.6	88.2	84.9	78.3	89.8	89.9	90.0	75.9	90.0	80.0	89.8	90.3	89.6	88.3	66.2	87.8	83.5	89.3	85.2

Table 6: Object detection results on the PASCAL VOC 2012 test set. All models use VGG-16 as the backbone network.

表 6： PASCAL VOC 212 测试机上的目标检测结果。所有的模型使用 VGG-16 作为主干网络。

Method	Data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
RefineDet320	07++12	78.1	90.4	84.1	79.8	66.8	56.1	83.1	82.7	90.7	61.7	82.4	63.8	89.4	86.9	85.9	85.7	53.3	84.3	73.1	87.4	73.9
RefineDet512	07++12	80.1	90.2	86.8	81.8	68.0	65.6	84.9	85.0	92.2	62.0	84.4	64.9	90.6	88.3	87.2	87.8	58.0	86.3	72.5	88.7	76.6
RefineDet320+	07++12	82.7	92.0	88.4	84.9	74.0	69.5	86.0	88.0	93.3	67.0	86.2	68.3	92.1	89.7	88.9	89.4	62.0	88.5	75.9	90.0	80.0
RefineDet512+	07++12	83.5	92.2	89.4	85.0	74.1	70.8	87.0	88.7	94.0	68.6	87.1	68.2	92.5	90.8	89.4	90.2	64.1	89.8	75.2	90.7	81.1
RefineDet320	COCO+07++12	82.7	93.1	88.2	83.6	74.4	65.1	87.1	87.1	93.7	67.4	86.1	69.4	91.5	90.6	91.4	89.4	59.6	87.9	78.1	91.1	80.0
RefineDet512	COCO+07++12	85.0	94.0	90.0	86.9	76.9	74.1	89.7	89.8	94.2	69.7	90.0	68.5	92.6	92.8	91.5	91.4	66.0	91.2	75.4	91.8	83.0
RefineDet320+	COCO+07++12	86.0	94.2	90.2	87.7	80.4	74.9	90.0	91.7	94.9	71.9	89.8	71.7	93.5	91.9	92.4	91.9	66.5	91.5	79.1	92.8	83.9
RefineDet512+	COCO+07++12	86.8	94.7	91.5	88.8	80.4	77.6	90.4	92.3	95.6	72.5	91.6	69.9	93.9	93.5	92.4	92.6	68.8	92.4	78.5	93.6	85.2

Table 7: Object detection results on the MS COCO test-dev set.

表 7： MS COCO test-dev 测试集上的目标检测结果。

Method	Net	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
RefineDet320	VGG-16	29.4	49.2	31.3	10.0	32.0	44.4	26.2	42.2	45.8	18.7	52.1	66.0
RefineDet512	VGG-16	33.0	54.5	35.5	16.3	36.3	44.3	28.3	46.4	50.6	29.3	55.5	66.0
RefineDet320	ResNet-101	32.0	51.4	34.2	10.5	34.7	50.4	28.0	44.0	47.6	20.2	53.0	69.8
RefineDet512	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4	30.6	49.0	53.0	30.0	58.2	70.3
RefineDet320+	VGG-16	35.2	56.1	37.7	19.5	37.2	47.0	30.1	49.6	57.4	36.2	62.3	72.6
RefineDet512+	VGG-16	37.6	58.7	40.8	22.7	40.3	48.3	31.4	52.4	61.3	41.6	65.8	75.4
RefineDet320+	ResNet-101	38.6	59.9	41.7	21.1	41.7	52.3	32.2	52.9	61.1	40.2	66.2	77.1
RefineDet512+	ResNet-101	41.8	62.9	45.7	25.6	45.1	54.1	34.0	56.3	65.5	46.2	70.2	79.8

Figure 3: Qualitative results of RefineDet512 on the PASCAL VOC 2007 test set (corresponding to 85.2% mAP). VGG-16 is used as the backbone network. The training data is 07+12+COCO.

图 3： RefineDet512 在 PASCAL VOC 2007 测试集上的定性检测结果（相应的 mAP 为 85.2%）。主干网络使用 VGG-16。训练数据为 07+12+COCO。

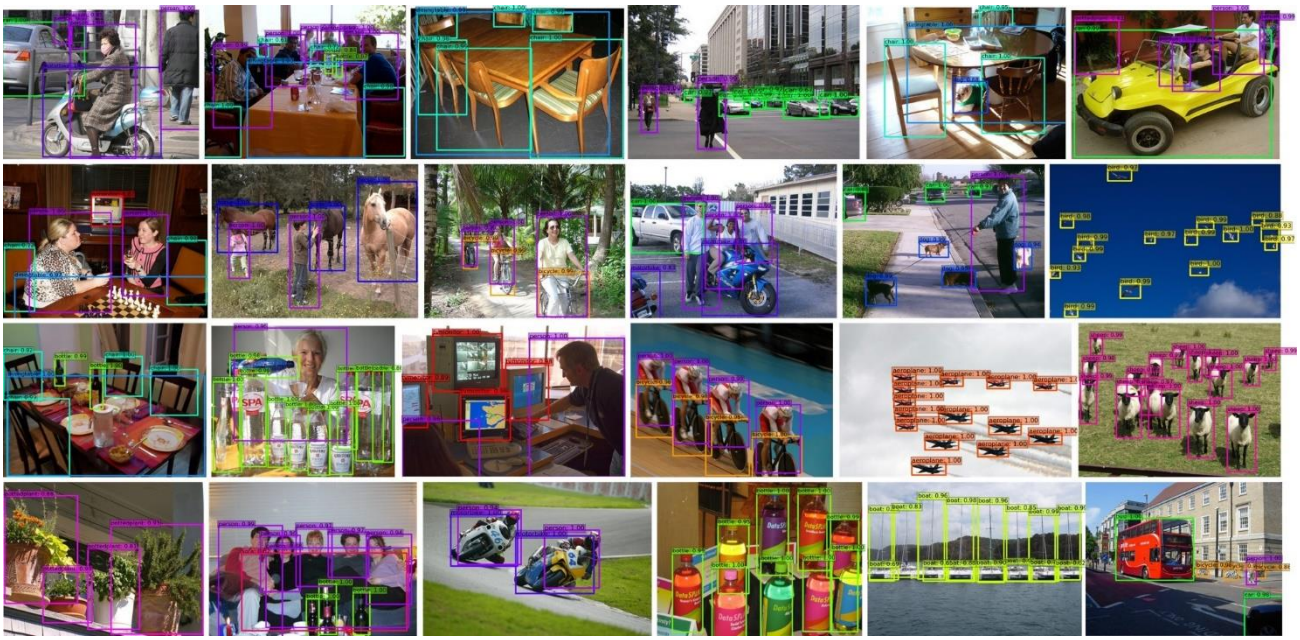

Figure 4: Qualitative results of RefineDet512 on the PASCAL VOC 2012 test set (corresponding to 85.0% mAP). VGG-16 is used as the backbone network. The training data is 07++12+COCO.

图 4： RefineDet512 在 PASCAL VOC 2012 测试集上的定性检测结果（相应的 mAP 为 85.0%）。主干网络使用 VGG-16。训练数据为 07+12+COCO。

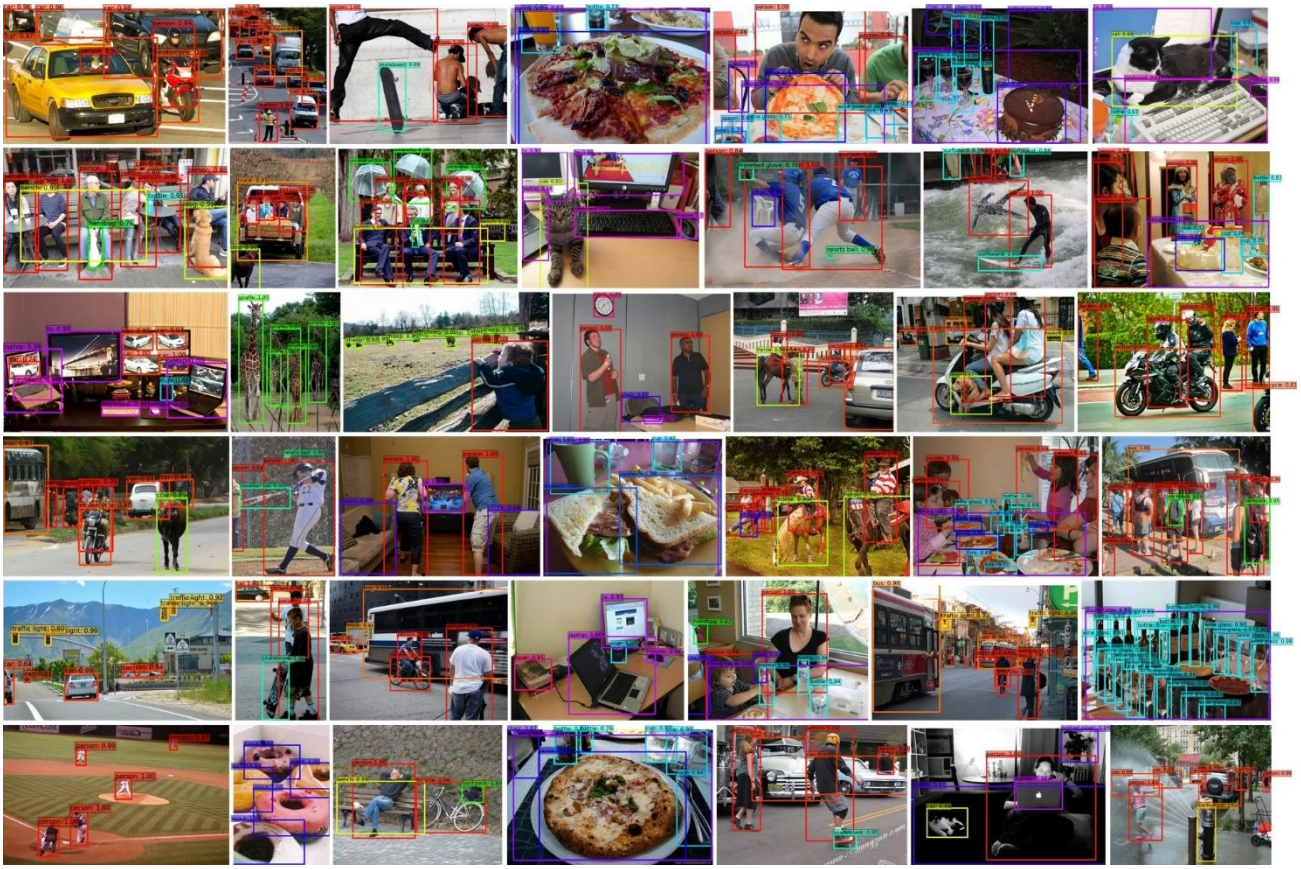

Figure 5: Qualitative results of RefineDet512 on the MS COCO test-dev set (corresponding to 36.4% mAP). ResNet-101 is used as the backbone network. The training data is COCO trainval35k.

图 5： RefineDet512 在 MS COCO test-dev 测试集上的定性检测结果（相应的 mAP 为 36.4%）。主干网络使用 ResNet-101。训练数据为 COCO trainval35k。

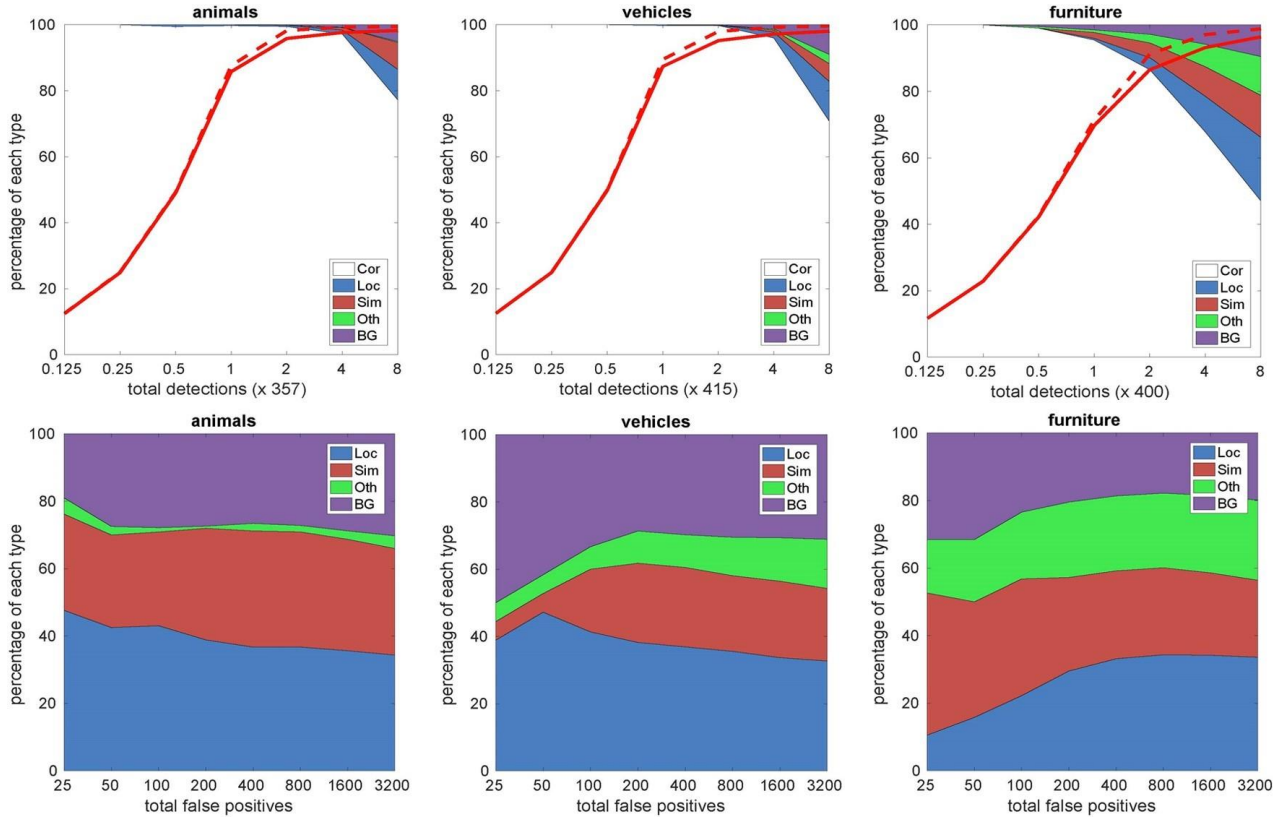

Figure 6: Visualization of the performance of RefineDet512 on animals, vehicles, and furniture classes in the VOC 2007 test set. The top row shows the cumulative fraction of detections that are correct (Cor) or false positive due to poor localization (Loc), confusion with similar categories (Sim), with others (Oth), or with background (BG). The solid red line reflects the change of recall with strong criteria (0.5 jaccard overlap) as the number of detections increases. The dashed red line is using the "weak" criteria (0.1 jaccard overlap). The bottom row shows the distribution of the top-ranked false positive types.

图 6：在 VOC 2007 测试集中对 RefineDet512 在动物，车辆和家具类上的表现进行可视化。 第一行显示由于不良定位（Loc），与类似类别（Sim）混淆，与其他人（Oth）或背景（BG）相关的正确检测（Cor）或假阳性的累积分数。 红色实线表示随着检测次数增加，具有较强标准（0.5 jaccard 重叠）的召回变化。 红色虚线使用"弱"标准（0.1 jaccard 重叠）。 最下面一行显示排名靠前的假阳性类型的分布。

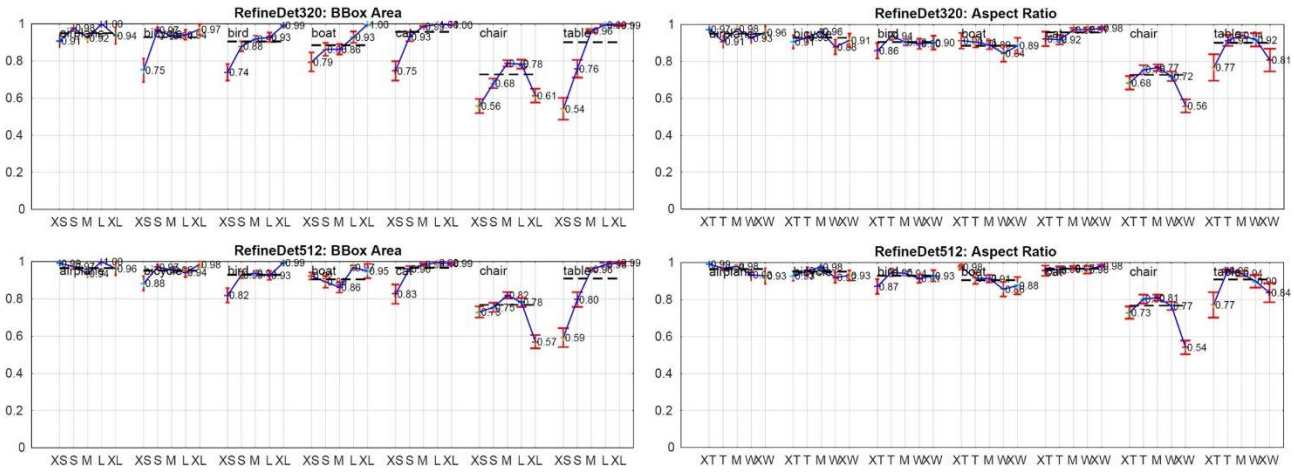

Figure 7: Sensitivity and impact of different object characteristics on the VOC 2007 test set. The plot on the left shows the effects of BBox Area per category, and the right plot shows the effect of Aspect Ratio. Key: BBox Area: XS=extra-small; S=small; M=medium; L=large; XL =extra-large. Aspect Ratio: XT=extra-tall/narrow; T=tall; M=medium; W=wide; XW =extra-wide.

图 7：不同目标特性对 VOC 2007 测试集的灵敏度和影响。 左侧的图表显示了每个类别的 BBox 面积的效果，右侧的图表显示了长宽比的影响。 关键：BBox 区域：XS =超小;S =小;M =中等;L =大; XL =超大。 长宽比：XT =超高/窄;T =高;M =中等;W =宽; XW =超宽。