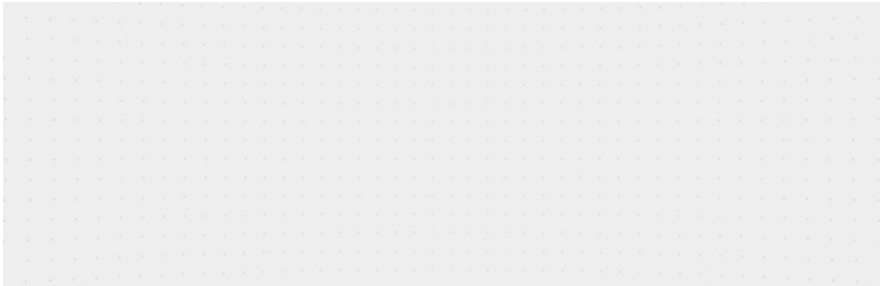


多领域实时目标检测算法最新综述

汽车人 极市平台 2022-09-16 22:00:06 发表于广东 手机阅读 罍

↑ 点击蓝字 关注极市平台



作者 | 汽车人

来源 | 自动驾驶之心

编辑 | 极市平台

极市导读

本文在广泛的数据集上对多个实时目标检测器进行了全面的研究 >>加入极市CV技术交流群，走在计算机视觉的最前沿

1 摘要

基于深度神经网络的目标检测器不断发展，并用于多种应用，每个应用都有自己的一组要求。安全关键型的应用程序需要高精度和可靠性，而低延迟的任务需要节约资源的神经网络结构。实时目标检测器在有高影响力的现实世界应用中是必不可少的，并不断有新方法提出，但它们过分强调精度和速度上的改进，而忽略了其他方面，如多功能性、鲁棒性、资源和能源效率。现有网络的参考基准不存在，新网络设计的标准评估指南也不存在，这导致比较模棱两可和不一致。因此，作者在大规模的数据集上对多个实时检测器（anchor-based、anchor-free和Transformer）进行了全面调研，并输出了一组指标结果。作者还研究了图像大小、anchor尺寸、置信度阈值和层结构等变量对整体性能的影响。作者分析了检测网络对分布变化、自然损坏和对抗性攻击的鲁棒性。此外，作者提供校准分析来衡量预测的可靠性。最后，为了突出现实世界的影响，作者针对自动驾驶和医疗保健应用进行了案例研究。为了进一步衡量网络在关键实时应用中的能力，作者测试了在边缘设备上部署检测网络后的性能。作者大量的实证研究可以作为工业界在现有网络上进行选择的指南。作者还希望激励研究界朝着网络设计和评估的新方向发展，专注于更大和整体的综述，以产生更深远的影响。

2 概述

深度神经网络的最新进展带来了目标检测领域的显著突破。目标检测通过提供目标的位置以及类别标签和置信度分数来同时完成分类和定位。目标检测器可用于多种应用场景，例如自动驾驶系统(ADS)、监控、机器人和医疗保健。ADS需要实时准确地检测车辆、交通标志和其他障碍物，此外，为了确保安全，它们需要检测器在不同的光照和天气条件下可靠且一致地执行。医疗保健应用需要高精度，即使它不是非常快。低延迟应用程序需要部署在边缘设备上，因此需要快速且紧凑的检测器以适应低功耗硬件设备。不同的应用程序有不同的标准，现实世界的设置伴随着时间和资源的限制。因此，检测器需要资源和能源效率高，以确保部署在高影响力的实际应用中。这需要对不同标准的实时检测网络进行详细分析。

目前人们已经提出了许多实时检测网络，它们实现了SOTA性能，但它们主要关注精度和速度，但省略了其他指标，如简单性、适用性和能源效率等。而这些指标，还包括泛化、鲁棒性和可靠性

壹伴图

极市平台
extreme

月发文数目: **
月平均阅读: **

文章工具

已发文

采集图文 合成多

采集样式 查看

等网络能力的评估都是很重要的。因此，作者进行了全面的、在不同数据集上、多个检测器上，进行实时目标检测以及目标检测基准的研究，作者还提供了两个关于自动驾驶和医疗保健应用的案例研究。除了精度和推理时间外，作者还评估每个模型的资源 and 能源消耗，以估计环境影响。作者选择了众多网络并创建了一个统一的框架，可以轻松分析骨干网（或特征提取器）和检测头的不同组合（见下表）。

Table 1: The summary of all the detection heads, backbones, datasets, and deployment hardware used for the experiments in this study.

Detection Head	Backbone	Dataset	Hardware
ThunderNet	ShuffleNet-v2; EfficientNet-B0	VOC; COCO	NVIDIA 2080Ti
YOLO; SSD	MobileNet-v2; DeiT-T	Corrupted COCO	Jetson Xavier
DETR	DarkNet-19; ResNet-18	BDD; Cityscapes	Jetson TX2
CenterNet; TTFNet; FCOS; NanoDet	Xception; HarDNet-68; VoVNet	Kvasir-SEG	

为了进一步详细评估性能增益/损失，作者解耦了不同变量的影响，例如图像大小、anchor大小、置信度阈值和架构层类型。为了进行统一的评估，作者遵循标准的实验流程并详述使用的所有参数。每个组合都在两个广泛使用的通用数据集（PASCALVOC(Everinghametal.,2010)和MSCOCO(Linetal.,2014))上进行训练和测试。网络应该对实时应用中不断变化的光照和天气条件具有鲁棒性，因此，作者进一步进行了大量的鲁棒性分析，以分析网络在分布变化和自然损坏方面的结果。对于安全关键型应用，网络也应该对抗抗性具有鲁棒性，包含人眼无法察觉的变化的图像，因此作者评估网络对此类攻击的鲁棒性。同样，对于这些应用程序，不确定性的度量有助于及时做出决策，因此作者还提供了每个网络的可靠性分析。最后，为了展示对现实世界的影响，作者对自动驾驶和医疗保健领域进行了两个独家案例研究。对于前者，将检测器性能移植到Berkeley Deep Drive(BDD)(Yuetal.,2018)数据集上，这与ADS应用更相关。作者还展示了分布外(OOD)数据集Cityscapes(Cordtsetal.,2016)的泛化能力和性能报告。为了突出检测器实时部署的可行性，作者在嵌入式硬件部署上使用NVIDIA TensorRT优化模型，并详述了低功耗设备上的实时性能。对于医疗保健案例研究，作者展示了网络从医学图像中检测息肉的能力，这些息肉用于检测患者是否得癌症。这些应用涵盖了两个具有不同要求的不同领域，作者的案例研究提供了超越标准的独特视角基准测试，并衡量检测器在实时应用中更相关和适用的不同数据集的能力。

作者制定了8种度量标准，即精度、对自然和对抗性破坏的鲁棒性、速度、参数量、MAC (Multiply-Accumulate operations)计数、能耗和校准误差(衡量可靠性)。如下图，理想的网络应该占据整个八边形，这样的网络具有最高的精度、鲁棒性和速度，参数量和MAC计数，同时消耗最低的能量，是被校准的最好的。唯一一个实时两阶段检测器ThunderNet是为移动设备设计的，在资源方面效率很高，但在准确性、自然鲁棒性方面不足，是最慢的网络之一。YOLO是一种基于anchor的检测器，其能量消耗排名第二，处于校准的中间范围，但在速度、精度和鲁棒性方面落后。SSD是另一种基于anchor的检测器，在精度和速度之间提供了很好的平衡。它具有最佳的校准评分，更可靠。DETR是一种基于transformer的检测器，它的MAC计数最低，在对抗鲁棒性方面排名第二，但它的校准分数最低，因此预测的可靠性较低。CenterNet对抗抗性攻击具有最高的鲁棒性，是第二快的，并且在所有其他指标上也处于良好的位置。TTFNet位于中间位置。FCOS具有最高的准确性和稳健性，但在其他指标上不稳定。NanoDet在速度方面是最快的，精度上是第二好的，并且资源消耗最低。这四种检测器都属于anchor-free的,是基于关键点的检测器范畴。总的来说，NanoDet在大多数顶点上都达到了最高点，并且在校准上获得了平均值，因此，对于需要在低功耗设备上运行、速度和精度高的应用程序来说，NanoDet是一个较好的选择。

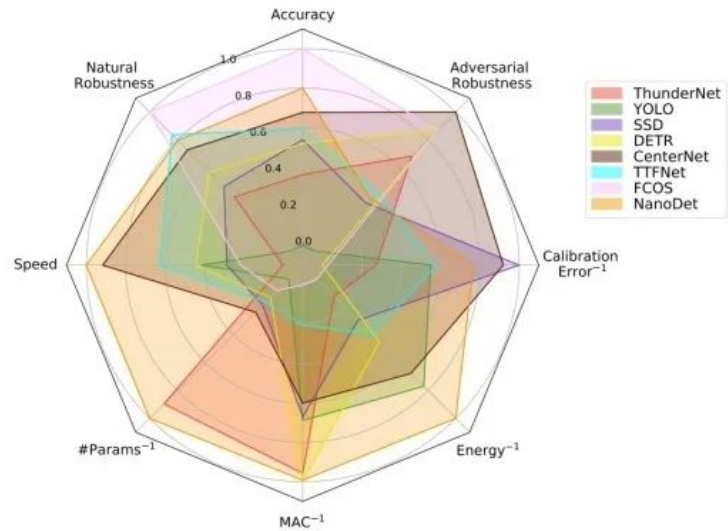


Figure 1: Overall performance of different object detector networks (with HarDNet-68 backbone) trained on COCO dataset on different metrics. All metrics are normalized w.r.t minimum and maximum while for the four metrics with ⁻¹ superscript, normalization was applied on the inverse values. Therefore, a network with full coverage of the octagon has all the ideal features: highest accuracy, natural/adversarial robustness, and speed with lowest number of parameters, MAC count, energy consumption, and calibration error (check Section A.1 for more details).

作者总结发现，基于关键点（anchor-free）的方法在精度和速度上普遍优于基于anchor的方法和两阶段方法。作者还注意到，虽然较高的MAC计数可能导致更高的能量消耗，但它们不一定会导致准确性的提高。所有的检测器对大中型目标的检测精度较高，但对小目标的检测精度较低。FCOS在与较重的骨干(如HarDNet-68)配对时，在检测小目标方面表现相对较好。增加输入图像尺寸不总是有益的，因为速度的下降往往超过精度的提高。anchor大小影响基于anchor检测器预测的不确定性使得它们难以适应较新的数据集。基于关键点的检测器对于跨多个数据集有较好的泛化能力。不同的置信度阈值在精度和速度上的变化显示了再现结果的模糊性。由于transformer使用注意力模块来捕捉全局信息，它们对不同的图像大小不那么敏感，并获取比较一致的性能。校准结果表明，基于关键点的方法是谨慎和不自信的，从而证明在安全关键应用中有用。但是，有趣的是，DETR是所有网络中最自负的。随着transformers获得越来越多的关注，这种详细的分析将为这种新的架构范式的功能和缺陷提供更多的见解。ADS的案例研究表明，在一台设备上看到的性能趋势并不一定会转化到部署中使用的嵌入式硬件。医疗保健案例研究表明，精度相对较高的网络可能没有更高的召回值，而这些召回值在医疗数据中更重要(因为假阴性比假阳性更有害)。

本文的贡献总结如下：对九个特征提取网络和八个检测头的组合进行了广泛的实证研究，范围从two-stage、one-stage、anchor-based、keypoint-based到transformer-based架构。

1. 结果详细，包括基准数据集上的精度、速度、可学习参数量、MAC计数和能耗。
2. 变量的影响，例如图像大小、anchor大小、置信度阈值和特定架构设计对整体性能的影响。
3. 针对15种不同的自然损坏和强度不同的对抗性攻击对所有网络进行鲁棒性分析。
4. 通过评估所有网络的校准分数进行可靠性分析。
5. 通过对更相关的BDD数据集进行分析，对自动驾驶系统进行案例研究。并且，通过在Cityscapes数据集上测试网络对分布外数据的泛化性能。
6. 在边缘设备上部署TensorRT优化检测器：Jetson-Xavier和Jetson-Tx2。
7. 通过对医疗保健应用进行分析的案例研究，用于检测癌性息肉的Kvasir-SEG数据集。

3 目标检测回顾

目标检测通过提供目标实例的类别标签和边界框坐标来同时进行分类和定位。基于卷积神经网络(CNN)的目标检测器通常分为两类，即两阶段和单阶段检测方法，详见下表。

Table 2: An overview of the details of the detection methods. The loss functions split up into classification (Cls) and localization (Loc) losses. Convolution layer types include vanilla 2D convolution (V), deformable convolution (DCN), bilinear upsampling (UP), depth-wise separable convolution (DWS), and transposed convolution (T).

Head	Localization Type		Multi-scale	Neck Type	Convolution Layers	Loss Functions		NMS Layer	
						Cls	Loc		
Thundernet	two-stage	anchor-based	✓	3	CEM	V	CE	reg: Smooth L1	Soft-NMS
Yolo	one-stage	anchor-based	✓	2	-	V	FL	reg: L2; conf: L2	NMS
SSD	one-stage	anchor-based	✓	6	FPN	V	CE	reg: Smooth L1	NMS
DETR	one-stage	self-attention	✗	-	-	-	CE	reg: L1 + GIoU	-
CenterNet	one-stage	keypoint-based	✗	-	-	V, DCN,T	FL	off: L1; emb: L1	Maxpool
TTFNet	one-stage	keypoint-based	✓	4	-	V, DCN, UP	FL	reg: GIoU	Maxpool
FCOS	one-stage	keypoint-based	✓	5	FPN	V	FL	reg: GIoU	NMS
Nanodet	one-stage	keypoint-based	✓	3	PAN	V, DWS	GFL	reg: GIoU	NMS

two-stage

两阶段检测器由一个单独的region proposal网络(RPN)进行前景背景分类。从RPN中提出的感兴趣区域(ROI)中提取的特征被传递给分类头以确定类标签，并传递给回归头以确定边界框位置(参考Faster RCNN系列)。基于区域的卷积神经网络(RCNN)使用选择性搜索算法来查找图像中可能是目标的像素区域，然后将Proposal输入CNN(Girshicketal.,2014)。从CNN中提取的特征支持向量机(SVM)进行分类和并回归边界框。RCNN需要渐进式多阶段训练，而且速度很慢。为了克服RCNN的缺点，Fast-RCNN提出了一些修改（Girshick，2015）。首先，不是提取selective search后的图像区域的特征，而是使用CNN直接提取整个图像的特征。然后使用一个ROI pooling层来得到和图像proposal区域对应的特征。其次，将SVM分类器和回归器分别替换为全连接层。Faster-RCNN提出了进一步的改进，以摆脱对速度较慢的Region proposal选择性搜索算法。由主干CNN提取的特征被发送到一个额外的基于CNN的region proposal网络(RPN)，该网络提供region proposal(Renetal.,2015)。然而，尽管精度很高，但上述两阶段检测方法并不适合实时应用。一种名为ThunderNet(Qinetal.,2019)的轻量级两阶段检测器，该检测器具有高效的RPN和用于实时检测的小型骨干网络。

one-stage

单阶段目标检测器由单个端到端前馈网络组成，整体执行分类和回归。这些检测器没有单独的proposal生成阶段，而是将图像上的所有位置视为潜在proposal。这些proposal中的每一个都用于预测类的概率、边界框位置和置信度分数。置信度分数决定了网络对其类别预测的确定程度。

单阶段检测器中的主要两类是anchor-based和anchor-free的检测器。anchor-based的检测器使用预定的anchor框（或先验）来辅助预测。这种方法的突出例子是You Only Look Once(YOLO;Redmonetal.(2016);Redmon&Farhadi(2017;2018))和Single Shot Detector (SSD;Liuetal.(2016))。YOLO的工作原理是将输入图像抽象为单元格网，其中每个单元格负责预测边界框（如果框的中心落在单元格内）。每个网格单元预测多个边界框并输出位置和类别标签以及它的置信度。SSD是第一个在保持实时速度的同时与当代两阶段检测器精度相匹配的单阶段检测器。SSD在FPN上的特征图上的每个位置，预测一组固定的但不同尺度anchor的目标置信度和目标框偏移量。FPN主要用于生成多分辨率特征(Linetal.,2017a)。

Anchor-based的检测器需要处理高度依赖数据集的anchor数量、纵横比和大小等超参数，这个缺点无法避免。这导致引入了anchor-free（又名基于关键点）目标检测器这种新范式。anchor-free的方法将目标视为点，而不是将它们建模为边界框。预测关键点，例如目标的角或中心，并且宽度和高度是从这些点而不是预定的anchor来回归。引入了几个基于关键点的网络，即CornerNet、CenterNet、FCOS、NanoDet和TTFNet（Law&Deng，2018；Zhou等人，2019a；Tian等人，2019；Lyu，2020；Liu等人，2020）。尽管anchor-based和anchor-free的检测器在通用目标检测中都取得了显著的精度，但它在很大程度上被缺乏全局上下文信息的基于CNN的架构所主导。此外，现代检测器通常对大量proposal、anchors或窗口中心执行回归和分类。因此，它们的性能受到复杂的后处理任务（例如NMS）的影响。

Vision transformers已被引入作为CNN的替代架构范式。基于Transformer的检测器，例如DETR(Carionetal.,2020)，利用自注意力模块显式地对给定序列中元素之间的所有交互进行建模，从

而提供全局上下文信息。Transformer的整体设计还通过对给定输入进行直接预测，绕过了NMS等手工操作过程。

4 目标检测配方

基础概念

目标检测问题可以形式化为：给定任意图像和预定义的目标类别列表，目标检测模型不仅对图像中存在的目标实例类型进行分类 $\{c_1, c_2, \dots, c_m\}$ ，还返回边界框形式的每个目标的位置 $\{b_1, b_2, \dots, b_m\}$ ，其中 $b_i = \{(x_1, y_1), (x_2, y_2)\}$ 是边界框的左上角和右下角坐标。目标检测器，包括单阶段和两阶段，通常由特征提取器（以下简称主干）和检测头组成。主干通常是基于CNN的网络，它提取场景中最突出的表示（从低级到高级特征）。大多数主干使用池化/卷积层来逐步减小特征图的大小并增加网络的感受野。然后将输出特征图传递给检测头，该检测头执行分类和回归以确定目标实例的标签和位置（下图显示了通用目标检测的组成）。

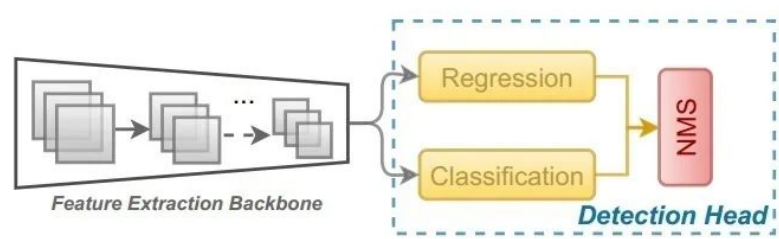


Figure 2: Schematic diagram of the main components of an object detector.

损失函数

作者简要概述了用于训练目标检测器的损失函数。两个目标函数通常用于训练基于CNN的检测器，即分类和回归损失。分类损失通常由交叉熵（CE）损失定义，如下：

$$\mathcal{L}_{CE} = - \sum_{i=1}^n t_i \log(p_i)$$

其中 t_i 是ground-truth标签， p_i 是第 i 类的softmax概率。然而，CE损失并没有考虑不平衡的数据集，与频繁出现的目标相比，不太频繁的目标更难学习。因此，研究人员(2017b)提出了Focal Loss(FL)，它通过对困难样本分配更多的权重，同时降低容易学习样本的损失贡献来解决类别平衡问题：

$$\mathcal{L}_{FL} = -\alpha_i(1 - p_i)^\gamma \log(p_i)$$

其中 α_i 是加权参数， $\gamma \geq 0$ 是可调制参数。回归损失通常是在ground-truth和预测边界框之间的所有四个边界框坐标上的L1（最小绝对偏差）或L2（最小二乘误差）损失。

Anchor-based 和 Keypoint-based

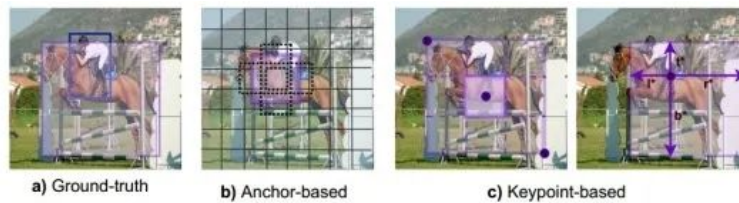


Figure 3: (b) Anchor-based techniques assign anchors to each grid in the image and make predictions as offsets to these. (c) The Keypoint-based techniques forego anchors and estimate keypoint, for instance, corners or center of object) onto heatmap and regress the width and height of the object from this keypoint.

anchor-based的目标检测技术使用anchor框的概念（在文献中也可称为先验框）。在这种方法中，图像被划分为网格，其中每个网格单元可以分配给多个预定义的anchor框（上图b）。这些框被定义为捕获特定目标类的比例和纵横比，通常根据训练数据集中的目标大小进行选择。计算anchors和ground-truth边界框之间的交集（IoU），具有最高重叠的anchor用于预测该目标的位置和类别。当anchor与ground-truth box的重叠较高且超过一个给定的阈值时，它被认为是一个正anchor。该网络不是直接预测边界框，而是预测平铺的anchor框的偏移量，并为每个框返回一组唯一的预测。anchor框的使用有助于检测多个、不同尺度的目标和重叠的目标。然而，anchor-based的方法有两个主要缺点。首先，需要大量的anchor框来确保与真实框有足够的重叠，而在实践中，只有一小部分与真实框重叠。这在正负anchor之间造成了巨大的不平衡，从而增加了训练时间。其次，anchor框的大小、形状和纵横比高度依赖于数据集，因此需要对每个数据集进行微调。然而，这些anchor是使用数据集的ground-truth框得出的，这在多尺度架构中变得更加复杂，其中每个尺度使用不同的特征和自己的一组anchor。为了缓解上述问题，提出了不使用anchor的anchor-free的目标检测技术（Law&Deng,2018;Zhouetal.,2019a;Tianetal.,2019）。检测问题被重新表述为逐像素预测，类似于分割。CNN的特征用于创建热力图，其中强度峰值代表关键点，例如相关目标的角点或中心点。除了这些，还有一些额外的分支可以预测外接框的尺寸（宽度和高度）。热力图预测与嵌入一起用于估计预测框的正确位置和大小。对于中心关键点，预测从中心到目标边界框四个边的距离以进行目标检测（上图c）。

NMS

目标检测器产生了太多的proposal，其中许多是多余的。为了去除密集的重复预测结果，检测器通常使用称为NMS的后处理步骤。NMS模块首先根据每个实例的置信度分数对预测的proposal进行排序，并选择具有最高置信度的proposal。随后，计算其它proposal分别于最高置信度的proposal执行IoU，公式如下：

$$IoU(b_m, b_i) = \frac{b_m \cap b_i}{b_m \cup b_i}$$

其中， b_m 是具有最高置信度的proposal， b_i 表示为真值生成的其它proposal。如果此值大于设置的NMS阈值（通常为0.5），则删除重复项。然而，与NMS相关的问题之一是，当proposals（针对不同实例）彼此接近或在某些情况下重叠时，有效的proposal会被抑制。对于拥挤场景尤其如此。因此，研究人员提出了Soft-NMS来改进NMS约束。在Soft-NMS中，与 b_m 重叠稍小的检测proposal（对于其他实例）的置信度分数衰减，同时确保与 b_m 重叠较高的proposal置信度分数衰减更多，从而可以去除重复项，又不至于完全抑制掉。这是通过计算（一个proposal b_i ）置信度得分和IoU与 b_m 的负值的乘积来完成的：

$$s_i = \begin{cases} s_i, & \text{if } IoU(b_m, b_i) < threshold \\ s_i(1 - IoU(b_m, b_i)), & \text{if } IoU(b_m, b_i) \geq threshold \end{cases}$$

anchor-free的检测器不使用这种基于IoU的NMS，因为它们处理的是热力图上的点而不是重叠

框。这些网络中的NMS是一个简单的基于峰值的maxpool操作，计算成本低。

目标检测中的挑战

目标检测作为计算机视觉问题本身就具有挑战性，因为理想的检测器必须以合理的能耗和计算成本提供高精度和高性能。稍后作者讨论了几种主干和检测头组合的优缺点，以展示精度和速度之间的权衡。检测精度和推理速度还取决于图像大小和目标大小。虽然通过从场景中提取更多信息，更高的图像分辨率会产生更好的准确性，但它也会降低推理速度。因此，选择在精度和速度之间提供适当平衡的图像尺寸至关重要。此外，目标大小在检测精度中起着重要作用。虽然检测器可以在大中型目标上实现高精度，但几乎所有检测器都难以检测场景中的较小目标 (Liu et al., 2021)。作者研究了目标大小和图像大小对检测精度和速度的影响。为了提供高精度，检测器需要具有鲁棒性，并对具有显著类内变化（例如，变化目标的大小、形状和类型）、姿势和非刚性变形。对于使用anchor-based的检测器，anchor的优化是一个挑战，因为它们依赖于数据集。后面作者展示了不同的anchor大小如何影响检测精度。另一个主要挑战是在不同的天气（雨、雪、暴风雪）和光照条件下保持一致的性能。对于自动驾驶等应用，检测器还必须考虑杂乱的背景、拥挤的场景和相机效果。作者在后面提供了关于检测器鲁棒性的详细研究。最后，深度神经网络倾向于依赖训练数据的监督进行快捷学习，因此过度拟合训练数据分布（分布内），而不是泛化到分布外(OOD)数据。真实场景的没见过的数据才是至关重要的，作者在后面提供了对分布内数据和分布外数据的详细分析。

5 目标检测中的Head

由于作者的研究范围是实时目标检测，作者专注于一个两阶段检测器：ThunderNet (Qin et al., 2019)，两个anchor-based的检测器：SSD (Liu et al., 2016)，YOLO (Redmon & Farhadi, 2017)，四个 (anchor-free) 基于关键点的检测器：CenterNet (Zhou et al., 2019a)、FCOS (Tian et al., 2019)、NanoDet (Lyu, 2020) 和TTFNet (Liu et al., 2020)，和一个基于Transformer的检测器：DETR (Carion et al., 2020)

ThunderNet

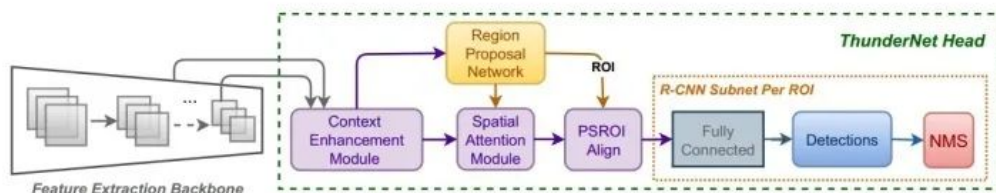


Figure 4: Schematic diagram of two-stage detector, ThunderNet.

ThunderNet重新审视了两阶段检测器架构并改进了Lighththead-RCNN(Liet al., 2017)，并使用ShuffleNet-v2(Ma et al., 2018)的变体作为主干。检测头在网络的早期阶段增加了通道数，以编码低级特征，这样提升了精度。ThunderNet使用了两个新模块：上下文增强模块(CEM)和空间注意力模块(SAM)。CEM聚合来自三个不同尺度的特征，通过利用局部和全局特征扩大感受野。SAM通过加强前景特征同时抑制背景特征来细化特征（上图所示）。SAM模块的输出为：

$$F_{SAM} = F_{CEM} \cdot \sigma(\mathcal{T}(F_{RPN}))$$

其中FSAM、FCEM和FRPN分别表示SAM、CEM和RPN模块的输出特征。 $\sigma(\cdot)$ 是sigmoid函数和 $\mathcal{T}(\cdot)$ 表示维度变换函数，以匹配来自FCEM和FRPN的输出通道数。

$$\mathcal{L}_{rpn} = \frac{1}{N_b} \sum_i \mathcal{L}_{cls}(p_i, t_i) + \lambda \frac{1}{N_a} \sum_i t_i \mathcal{L}_{reg}(b_i, b_g)$$

其中 \mathcal{L}_{cls} 是两个类（目标或非目标）.bi和bg分别表示第i个anchor的预测框和对应ground-truth目标框。与任何高于给定阈值的ground-truth重叠的anchor被认为是正样本（ $t_i=1$ ），其余的anchor被认为是负的（ $t_i=0$ ）。因此，乘法项确保回归损失仅对正anchor激活。 N_a 和 N_b 表示anchor位置的数量和batch大小， λ 是平衡权重。与FastR-CNN类似，执行ROI pooling并将这些区域发送到两个分支进行分类和回归，目标函数如下：

$$\mathcal{L} = \mathcal{L}_{cls}(p, u) + \lambda[u \geq 1]\mathcal{L}_{reg}(b_u, b_g)$$

其中 \mathcal{L}_{cls} 是真实类 u 的对数损失， λ 是平衡权重。 \mathcal{L}_{reg} 计算类 u 的ground-truth目标框和预测框的回归损失。 $[u \geq 1]$ 是逆向指标函数，当 $u \geq 1$ ($u=0$ 是背景类)。

YOLO

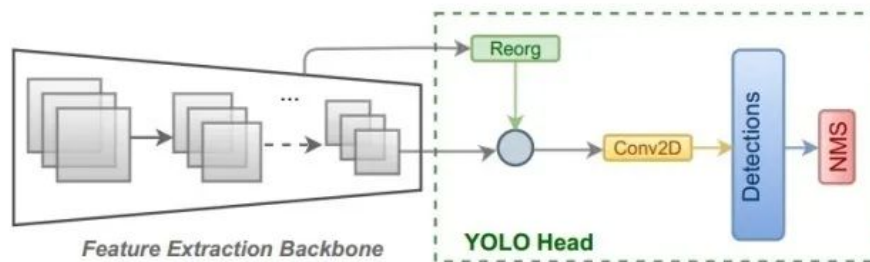


Figure 5: Schematic diagram of single-stage anchor-based detectors, YOLO.

YOLO(Redmon et al., 2016)是一个针对实时任务的单阶段目标检测网络。YOLO将图像划分为网格单元，每个单元预测一个由边界框和置信度分数。如果一个目标的中心位于某个特定的网格单元中，则称该目标属于该网格单元。YOLO快速简单，但召回率低。Redmon&Farhadi(2017)提出了YOLOv2来提高YOLO的精度和速度。YOLOv2不是对边界框进行任意预测，而是在每个网格中使用不同大小和纵横比的anchor来覆盖整个图像的不同位置 and 不同尺度。通过在特定数据集上使用基于IoU的k-Means聚类计算anchor大小，可以使anchor变得更准确。网络预测是每个anchor框的偏移量。YOLOv2在合并不同尺度的特征图获得的单个特征图上进行边界框预测（如上图）。

其它YOLO都是建立在YOLOv2基本概念之上，但有许多技巧和窍门来实现更高的性能。由于作者试图在简单的框架上进行评估，因此在本研究中，作者仅考虑YOLOv2版本，因为它简单、快速且具有最少的技巧。损失函数由分类损失、定位损失、和置信度损失（判断bbox是目标还是背景）：

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{reg} + \lambda' \mathcal{L}_{conf}$$

其中 \mathcal{L}_{cls} 是Focal Loss， \mathcal{L}_{reg} 和 \mathcal{L}_{conf} 都是L2 loss。 \mathcal{L}_{conf} 是衡量bbox是否为目标置信度损失（例如，如果一个框其实是背景，则其目标的置信度将降低）， λ 和 λ' 是平衡权重。

SSD

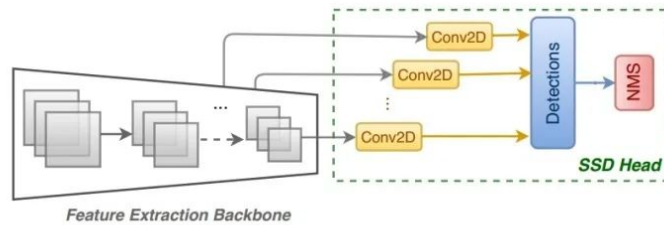


Figure 6: Schematic diagram of single-stage anchor-based detector, SSD.

SSD(Liuetal.,2016)有一个前馈CNN，它为场景中的多个目标实例生成边界框、置信度分数和分类标签。SSD使用多个特征图从逐渐降低的分辨率模拟不同大小的输入图像，同时跨尺度共享计算。浅层的特征图用于学习较小目标的低级特征，而较深层的特征用于定位场景中较大的目标。检测头为每个尺度的特征图采用单独的预定义anchor，最后结合所有预设anchor在不同尺度和纵横比下的预测结果。每个特征图的anchor的尺度和大小定义为：

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1} (k - 1)$$

其中 $k \in [1, m]$ ， s_{min} 和 s_{max} 的默认值分别为0.2和0.9。SSD中用了 $m=6$ 个特征图。SSD产生一组不同预测结果，涵盖各种形状和大小的目标。SSD使用匹配策略来确定哪些anchor对应于ground-truth，而与ground-truth最高重叠的那个anchor用于预测该目标的位置和类别。目标函数源自多目标 (Heetal.,2015)，并扩展到多个类别。总体目标函数是

$$\mathcal{L} = \frac{1}{N_{pos}} \mathcal{L}_{cls} + \lambda \mathcal{L}_{reg}$$

其中 \mathcal{L}_{cls} 是交叉熵损失， \mathcal{L}_{reg} 是所有与ground-truth匹配的正样本框的SmoothL1损失之和。 N 是正样本的数量， λ 是平衡权重。

CenterNet

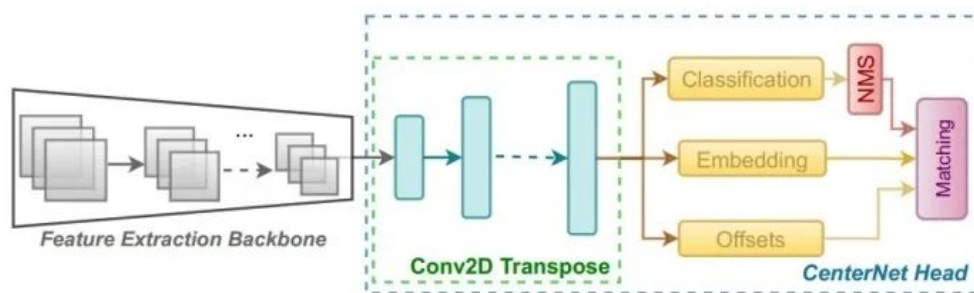


Figure 7: Schematic diagram of single-stage keypoint-based detector, CenterNet.

anchor-based的检测器必须处理与数据集高度相关的超参数，例如anchor的数量、纵横比和大小。CornerNet被提出作为anchor-based的方法的第一个替代方法，该方法将目标检测问题简化为关键点估计问题(Law&Deng,2018)。在(Lawetal.,2019;Zhouetal.,2019b)提出的多种方法中；a)，作者使用CenterNet(Zhouetal.,2019a)，因为它不仅比CornerNet获得更高的精度，而且还简化了关键点估计。检测算法通过三个转置卷积层来增强主干，以产生高分辨率输出。第一个分支输出一个热力图来估计目标的关键点或中心点，热力图的数量等于目标类别的数量。Ground-truth热力图是通过在ground-truth box的中心使用高斯核来创建的。峰值用于估计实例目标的中心并确定实例目标的类别。还有两个生成热力图的分支：embedding分支回归目标框的尺寸，即宽度和高度，offsets分支解释了将中心坐标映射到原始输入维度引起的离散化误差。总体目标函数给出为：

$$\mathcal{L} = \frac{1}{N_{pos}} \mathcal{L}_{cls} + \lambda \mathcal{L}_{reg}$$

其中 \mathcal{L}_{cls} 是使用FocalLoss减少像素级逻辑回归的惩罚(Linetal.,2017b), \mathcal{L}_{reg} 是L1损失, 以最小化中心坐标, 最后 \mathcal{L}_{mbis} 也是一个L1损失, 以最大限度地减少计算预测框的宽度和高度时的错误, λ 和 λ' 是平衡权重。

FCOS

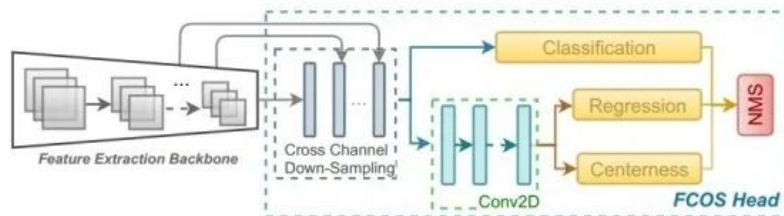


Figure 8: Schematic diagram of fully convolutional single-stage keypoint-based detector, FCOS.

FCOS是一种全卷积的anchor-free检测器, 将目标检测重新表述为类似于语义分割的逐像素预测问题(Tian等人, 2019)。检测器使用FPN的多级预测(Linetal.,2017a)来提高召回率并解决重叠边界框歧义。以不同的尺度获得五个特征图, 并在每个层上执行逐像素回归。这会增加召回率, 但会在远离目标中心的位置产生低质量的预测。为了避免这种情况, 并行添加了一个额外的分支, 以预测位置的中心位置。整体损失函数为:

$$\mathcal{L} = \frac{1}{N_{pos}} \mathcal{L}_{cls} + \frac{\lambda}{N_{pos}} \mathcal{L}_{reg} + \mathcal{L}_{cent}$$

其中 \mathcal{L}_{cls} 是FocalLoss, \mathcal{L}_{reg} 是IoU回归损失, \mathcal{L}_{cent} 是使用二元交叉熵(BCE) loss的中心损失。N是正样本的数量, λ 是平衡权重。IoU回归基于UnitBox(Yuetal.,2016), 是输入为IoU值的交叉熵损失的一种形式。与独立优化坐标值的L2 loss不同, IoU loss将其视为一个单元。最终的目标分数是由centerness得分加权得到。因此, 这个分支会降低距离目标中心较远的预测框的分数, 这有助于最终的NMS过滤掉低质量的预测结果。

NanoDet

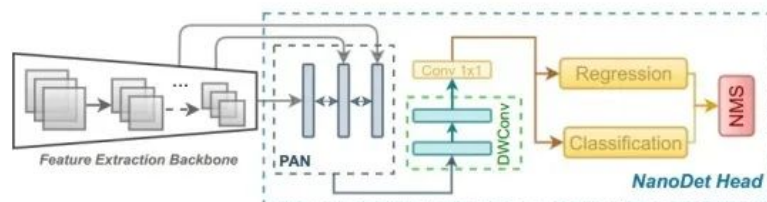


Figure 9: Schematic diagram of fully convolutional single-stage keypoint-based detector, NanoDet.

受FCOS的启发, NanoDet作为一种轻量级的anchor-free检测器被提出(Lyu, 2020)。Nano Det使用ATSS模块(Zhangetal.,2020), 该模块根据目标特征自动选择正负训练样本。检测器使用Generalized Focal Loss(GFL)(Lietal.,2020)进行分类和回归。GFL旨在将FocalLoss从离散域扩展到连续域, 以实现更好的优化。这是 Quality Focal Loss(QFL)和Distributed Focal Loss (DFL)的组合。QFL将分类置信度和IoU质量相结合, 最终输出一个分数, DFL将预测框视为连续分布并对其进行优化。Generalized IoU loss(GIoU)对于非重叠情况很有用, 因为它通过缓慢地向目标框移动来增加预测框的大小以与目标框重叠。用于训练NanoDetis的整体损失函数为:

$$\mathcal{L} = \frac{1}{N_{pos}} \sum_z \mathcal{L}_{QFL} + \frac{1}{N_{pos}} \sum_z 1_{\{c_z^* > 0\}} (\lambda \mathcal{L}_{GIoU} + \lambda' \mathcal{L}_{DFL})$$

其中LQFL和LDFL是QFL和DFL，LGIoU是GIoU损失。Npos是正样本的数量，λ和λ'是平衡权重。z表示金字塔特征图上的所有位置。FCOS使用五个特征图传递给多级FPN，而NanoDet使用三个特征图传递给三个单独的路径聚合网络（PAN）(Liu et al., 2018b)块。PAN类似于FPN，但通过添加自下而上的路径来增强较低级别特征。PAN块的输出连接到单独的检测头，这些检测头计算特定特征图的分类标签和边界框。NanoDet还删除了FCOS中的centerness分支，因此使其成为更快的变体。三个头的输出最终传递给NMS以实现输入图像的最终目标框和分类标签的预测。

DETR

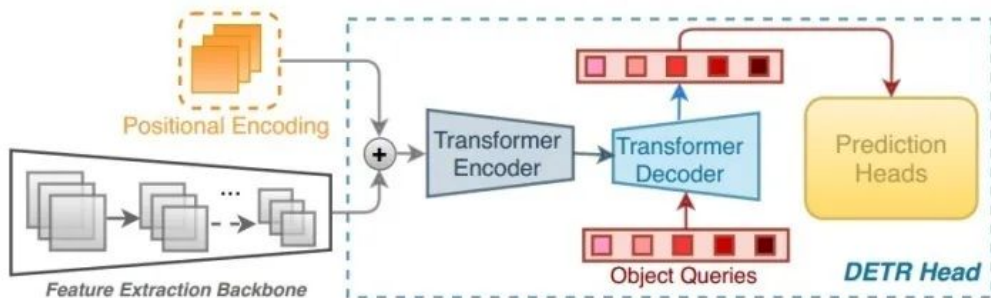


Figure 10: Schematic diagram of transformer-based detector, DETR.

Transformer是计算机视觉中一种新的设计范式，它依赖于注意力机制，并首次被DETR引入目标检测中（Carion等人，2020年）。DETR将目标检测任务转换为集合预测问题，消除了重复的边界框预测。Transformers通过使用自注意力模块基于整个图像上下文捕获目标之间的成对关系，从而避免重复预测。与使用NMS等后处理步骤来消除重复预测的传统目标检测器相比，有减少计算成本的优势。DETR由编码器-解码器转换器和进行最终预测的前馈网络(FFN)组成（上图）。编码器由一个多头自注意力(MHSA)模块(Vaswani et al., 2017)和一个FFN组成。这些块是排列不变的，因此，固定位置编码被添加到每个注意力层的输入中。解码器使用编码器特征并使用多个MHSA模块将目标查询转换为输出嵌入。N个输出嵌入被两个不同的FFN层使用，一个用于预测类标签，另一个用于预测框坐标。DETR使用唯一的二分匹配为每个给定的ground-truth找到最佳预测框。使用匈牙利优化算法有效地计算每N个查询到每N个ground-truth的一对一映射。在获得该集合的所有匹配对后，使用标准交叉熵损失进行分类，并使用L1和GIoU损失的线性组合进行回归。在每个解码器层之后添加辅助损失，以帮助模型在每个类中输出正确数量的目标。给定λ和λ'是平衡权重，总损失如下：

$$\mathcal{L} = \lambda \mathcal{L}_{cls} + \lambda' \mathcal{L}_{reg}$$

TTFNet

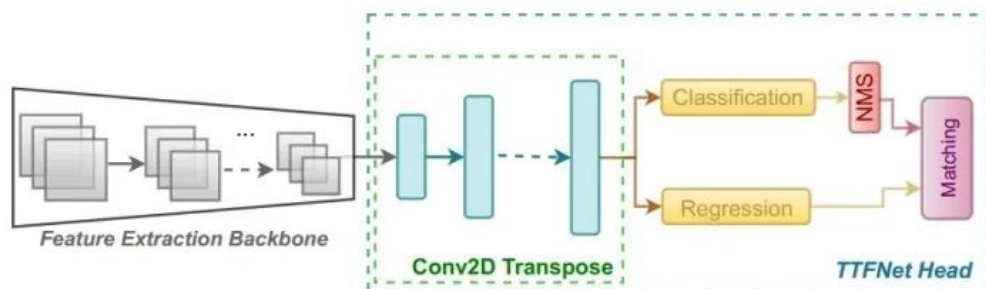


Figure 11: Schematic diagram of single-stage keypoint-based detector, TTFNet.

受CenterNet(Zhouetal.,2019a)的启发，TTFNet使用相同的策略，其中检测被视为中心定位和边界框尺寸回归的两部分问题(Liuetal.,2020)。对于中心定位，TTFNet采用高斯核在目标中心附近产生激活较高的热力图，类似于CenterNet，但另外还考虑了边界框的纵横比。对于尺寸回归，TTFNet提出将高斯区域中的所有像素作为训练样本，而不是只选择中心像素作为训练样本。此外，这些样本通过目标大小和高斯概率计算的权重进行加权，从而利用更多信息。这样做的原因是更多的训练样本类似于增加批量大小，这有助于扩大学习率并加快训练过程。TTFNet通过围绕目标中心构建一个子区域并仅从中提取训练样本来修改高斯核区域（详见上图）。使用高斯概率作为权重，以重点关注靠近目标中心的样本，从而减轻重叠歧义。由于目标尺寸的巨大差异，较大的目标比较小的目标产生更多的样本，因此较小目标的损失贡献可以忽略不计，这会影响检测精度。因此，引入了一种损失平衡策略，该策略充分利用大目标中的更多注释信息，同时保留较小目标的信息。

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_{(i,j) \in A_m} GIoU(\hat{b}_{ij}, b_m) \times W_{ij}$$

知乎 @自动驾驶之心

其中，真值 b_m ，采用高斯核，将子区域内的每个像素 A_m 视为回归样本。 \hat{b}_{ij} 是预测框， W_{ij} 是平衡权重， N_i 是回归样本数。因此，整体损失如下：

$$\mathcal{L} = \lambda \mathcal{L}_{cls} + \lambda' \mathcal{L}_{reg}$$

其中 $\lambda=1.0$ 和 $\lambda'=5.0$ 是分类和回归平衡权重， \mathcal{L}_{cls} 是Kong等人提出的FocalLoss的修改版本。(2019)

6Backbones

在这项研究中，作者根据速度、能耗和内存效率等因素选择了九个特征提取器作为主干，专门针对实时应用。在下文中，作者按时间顺序介绍了主干网络。

ResNet：Heetal。（2016）将网络层重新定义为具有残差跳跃连接的学习残差函数。具有跳跃连接的网络更容易优化，并且可以在增加深度的情况下获得相当大的精度。ResNet-18是深度残差网络的轻量级变体，由四个残差块组成，每个残差块有两个卷积，然后是BN层。

DarkNet：Redmon&Farhadi(2017)提出了一种计算轻量级的特征提取器DarkNet作为他们提出的实时目标检测算法YOLO的一部分。Darknet通过减少参数量对VGG-16进行了改进。出于实时检测的目的，本研究仅考虑了DarkNet-19。

Xception：Chollet(2017)提出的Xception作为对 Inception-V3的改进，完全基于深度可分离卷积(DWS;Kaiser等人（2017年））。所提出的架构是一个由36个深度可分离卷积层组成的线性堆叠，结构为14个模块，除了第一个和最后一个之外，所有模块都有残差连接。

MobileNet：Sandler等人（2018年）将MobileNet-v2设计为轻量级骨干网，专门用于嵌入式设备上的实时目标检测。该架构使用具有线性瓶颈和深度可分离卷积的反向残差块。它被称为倒置，因为在网络的狭窄部分之间存在跳跃连接，导致参数数量较少。此外，该网络包含跳过连接，以实现输入和输出瓶颈之间的特征可重用性。

ShuffleNet-v2：Ma等人(2018)设计了ShuffleNet-v2，通过降低内存访问成本来优化推理延迟。该架构的构建块由通道拆分操作组成，该操作将输入分成两部分，每一个都前馈到一个残差块。引入了通道混洗操作以实现两个拆分之间的信息传输以提高准确性。每个构建块的高效率使得可以使用更多的特征通道和更大容量的网络。

VoVNet：Leeetal(2019)提出VoVNet作为能耗低的实时检测的主干网络。它是使用One-Shot Aggregation(OSA)模块构建的，该模块仅在最后一个特征图中将所有中间特征连接一次。OSA块

中的卷积层具有相同的输入/输出通道，从而最大限度地减少了MAC计数，从而提高了GPU计算效率。本研究使用了速度更快、更节能的变体VoVNet-39。

EfficientNet: Tan&Le(2019)设计了EfficientNet，一种特征提取器，使用针对精度和MAC计数进行优化的自动多目标结构搜索算法。所提出的架构通过重新调整和平衡网络深度、宽度和分辨率来实现高精度。该架构的构建块使用Mobile Inverted Bottleneck Convolutions(MBConv)，还包括Squeeze 和 Excitation(SE)模块 (Hueta1.,2018) 。在提出的几个版本中，作者使用了EfficientNet-B0，该架构中最轻量级的版本。

HarDNet: Chaoetal (2019) 提出谐波密集连接网络 (HarDNet) 以在MAC计数和内存访问方面实现高效率。在减少DRAM (动态随机存取存储器) 方面，HarDNet在所有其他主干中脱颖而出。稀疏化方案提出了层之间的连接模式，使其类似于二次谐波的幂 (因此得名) 的重叠。所提出的连接模式形成了一组称为谐波密集块 (HDB) 的层，而不是考虑所有层，HDB中的梯度只有“logL”层。HDB的输出是L层以及之前所有奇数层的concat，一旦HDB完成，偶数层的输出将被丢弃。此外，使用stride-8代替stride-16 (在许多CNN网络中采用) 来增强局部特征提取。除了减少特征图的访问外，它还提供了其他优势，例如低延迟、更高的精度和更快的速度。在提出的几个版本中，HarDNet-68用于本研究。

DeiT:Touvronetal (2021) 修改了视觉Transformer以用作密集预测任务的特征提取器。所提出的架构，数据高效的图像Transformer(DeiT)，由重复的自注意力模块、前馈层和一个额外的蒸馏模块组成。为了提取有意义的图像表示，将来自最终Transformer块的学习嵌入发送到一个额外的模块，以在将其发送到检测头之前获得不同尺度的特征。该架构的最小版本，即DeiT-T (其中T代表tiny) 用于本研究。

7 验证评价

数据集

PASCAL VOC，以下称为VOC (Everingham等人，2010)，由20个目标类别组成，分为两个数据集，即VOC 2007和VOC 2012，共有21,493张图像，包含52,090个标注。

COCO(Linetal.,2014)是一个更具挑战性的数据集，由80个目标类别组成。作者使用包含118,287个图像 (860,001个标记实例) 的数据集的2017年拆分进行训练。

BDD (Yu等人，2018年) 是最大和最具挑战性的自动驾驶数据集之一。它包含各种驾驶场景，包括城市、高速公路和农村地区，以及代表现实驾驶挑战的各种天气和昼/夜驾驶条件。训练集包含约128万个标记实例的69,863张图像，测试集包含10个目标类别的10,000张图像和185,526个标注实例。

Cityscapes (Cordtsetal.,2016) 是一个记录完整的数据集，用于城市场景。作者从实例分割中提取边界框，然后将标记的注释分组为10个超类别以匹配BDD的类别。对于OOD评估，作者使用了500张图像和15949个边界框的测试集。

COCO有损，为了测试模型的稳健性，作者创建了一个数据集，通过添加损坏到原始COCO数据集来模拟在现实世界场景中发现的不同外部影响 (Michaelis等人，2019)。有15种不同的损坏，作者将它们分为四组：噪声、模糊、天气和数字化影响。噪声包括高斯噪声、脉冲噪声和散粒噪声。模糊包括散焦、透明度、运动和变焦模糊效果。作者使用亮度、雾、霜和雪来模拟不同的天气条件。最后，作者通过添加对比度、弹性变换、JPEG压缩和像素化的变化来解释数字化影响。这15种损坏适用于5种不同的严重程度。严重性级别范围从1 (不太严重的损坏) 到5 (最严重的损坏) 。

Kvasir-SEG (Jha等人，2020) 是用于定位胃肠道息肉的生物医学数据集。该数据集由1000张图像组成，每张图像中都存在息肉的分割掩码。该数据集还具有从分割掩码获得的边界框。这里，数据集分为800张图像用于训练和200张图像用于测试。

评价指标

目标检测器根据边界框和类别标签进行预测。在这里，作者首先通过计算IoU来测量预测边界框与ground-truth之间的重叠。基于IoU阈值，预测框分为真正例（TP）、假正例（FP），或假反例（FN）。接下来，作者计算精度和召回率：

$$precision = \frac{TP}{TP + FP},$$

$$recall = \frac{TP}{TP + FN}.$$

Precision衡量预测的精度，而召回率则显示模型找到所有正例的能力。高精度但低召回率意味着更多的FN（漏检），而相反则意味着更多的FP（误检）。精确召回(PR)曲线显示了不同阈值的精确度和召回值之间的权衡。

PR曲线向下倾斜，因为随着阈值的降低，会做出更多的预测（高召回率），而它们的精确度会降低（低精度）。作者计算各种IoU阈值下所有召回值（0到1之间）的平均精度（AP），这可以解释为PR曲线下的面积。最后，mAP（平均平均精度）是通过对所有类的AP进行平均来计算的。PASCAL VOC（Everingham等人，2010）以0.5IoU阈值（@IoU: 0.5）评估mAP，而COCO（Lin等人，2014）使用0.05步设置十个不同的阈值@IoU: [0.5-0.95]尺寸。在医疗保健等一些应用中，召回度量具有更大的价值，因为拥有更多的FN比FP更有害。平均召回是通过对所有IoU进行平均召回来衡量的，并且这些平均值被称为mAR。

作者还使用了F1分数指标，它衡量精度和召回率之间的平衡。F1分数计算如下：

$$F1 = 2 \times \left(\frac{precision \times recall}{precision + recall} \right)$$

作者计算检测器的卷积层、BN层和全连接层的MAC（乘法累加操作）计数，并得到可学习参数的数量（以百万为单位）。作者还得到了每个主干和检测头组合的每秒帧数(FPS)的推理速度。推理速度是针对500张图像计算得到的，并取平均值以消除偏差。最后，考虑到最近的节能AI趋势（Schwartz等人，2019），作者在整个测试数据集上计算模型的推理能耗。作者使用NVIDIA Management Library（NVIDIA，2019）来计算GPU在推理过程中的近似功耗。数据集的推理能耗以千焦(KJ)为单位显示，不包括其他组件的功耗。

实验设置

作者的完整框架在PyTorch 1.7(Paszkeetal.,2019)中实现，包括执行所有训练和评估的所有主干网络和检测头。需要注意的是，一些检测头（例如YOLO和DETR）在其原始实现中使用了多尺度训练，但是对于统一的训练方案，作者使用图像大小为512的单尺度训练。所有图像首先进行归一化，通过使用ImageNet均值(Russakovskyetal.,2015)，数据集的每个通道减去该均值。对于检测头，使用默认的PyTorch权重初始化（具有固定的种子值），使用ImageNet预训练的权重用于主干网络。

对于数据增强，作者使用expand、随机水平翻转、随机裁剪和随机光照，其中包括[0.5,1.5]范围内的随机对比度、饱和度[0.5,1.5]和色调[-18,+18]。作者使用batchsize=32,并使用随机梯度下降(SGD)优化器(Bottou,2010)训练模型，动量为0.9，学习率衰减因子为0.1。选择学习率调度器以确保所有模型的收敛。该规则的唯一例外是DETR，作者跟随作者并使用AdamW(Loshchilov&Hutter,2017)优化器。NMS阈值设置为0.45，置信度阈值设置为0.01。对于所有实验，作者在NVIDIA RTX 2080Ti GPU上评估模型。Pytorch模型使用NVIDIA TensorRT（8.0版）转换为其优化的高性能推理模型，以促进嵌入式硬件的部署。TensorRT转换通过融合网络中的多个层（包括

卷积和BN层）来优化网络，以实现并行处理。推理能耗是使用NVIDIA NVML API（Corporation, 2020）在单机上运行得到的。

8 结果

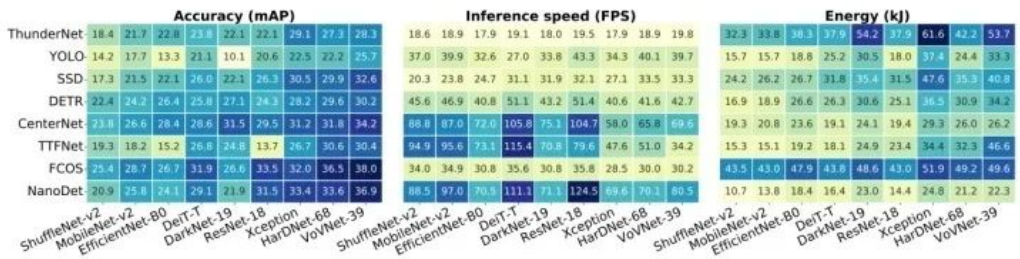


Figure 12: Summary result of all detection models with various backbones on the COCO dataset reported on three evaluation metrics - accuracy, speed, and energy. The darker shade and lighter shade in energy represents an ideal case.

上图展示了三个指标的总趋势：COCO数据集上的推理准确性、速度和能耗。以“精度”为关键指标对主干和检测头进行排序，并将结果从低到高排序。为了更容易进行性能分析，作者将主干和头分为三个频谱，即低、中和高，其中低和高是最不准确和最准确的主干和头组合，中间频谱包含具有平均性能的网络。

精度：从主干网络来看，VoVNet-39、HardNet-68和Xception在所有头上始终保持高精度，属于高频谱主干。VoVNet和HardNet的准确性可以分别归功于One-Shot Aggregation(OSA)模块和局部特征增强模块，而36个CNN层的线性堆栈有助于Xception。中谱被ResNet-18、DarkNet-19和DeiT-T占据。ResNet和DarkNet都是轻量级架构，而DeiT享有自注意力模块的好处，有助于利用全局信息。最后，EfficientNet-B0、MobileNet-v2和ShuffleNet-v2这些主要为减少MAC计数而设计的网络，精度最低。

对于检测头，处于精度高频谱的基本都是anchor-free的方法，比如NanoDet,FCOS,TTFNet和CenterNet等。它们没有根据目标大小定义anchor大小的麻烦。FCOS的FPN执行多尺度预测，并有一个中心分支来过滤低质量的预测。NanoDet使用PAN增强低级特征并使用有助于优化位置的GIoU损失。TTFNet和CenterNet还包含多个分辨率并进一步优化目标框定位。DETR中的注意力模块提高了精度，但由于主干仍然是CNN，因此性能掉到了中间频谱。SSD也占据了中间频谱，较低的频谱由YOLO和ThunderNet组成，它们分别是anchor-based的检测器和两阶段检测器。

速度：推理速度与精度相比呈现出不同的趋势。在主干网络中，“精度”位于中间频谱的网络最快，即ResNet-18、DarkNet-19和DeiT-T，从而促进准确性和速度之间的良好平衡。尽管ShuffleNet-v2是最不准确的一种，但由于其为低延迟而设计的架构，推理速度相当高。准确度最高的VoVNet-39和HardNet-68在速度方面位于中等频谱范围。然而，Xception是最慢的之一，因为它有大量的线性卷积层。

在检测头中，CenterNet、TTFNet和NanoDet是最快的，并且比其他检测器有很大的优势。CenterNet和TTFNet没有NMS瓶颈（因为它使用基于热力图峰值的max-pooling NMS而不是基于IoU的NMS），这有助于提高推理速度。FCOS，具有最高的精度，但在速度方面处于最低频谱，因为它具有五个特征图和一个额外的中心分支的重型架构。NanoDet类似于FCOS，但具有更轻量级的架构，只有三个特征图并且没有单独的分支，从而提高了推理速度。DETR在这里处于中间频谱，因为Transformer架构没有像CNN那样进行硬件优化（Ivanov等人，2020）。SSD和YOLO也位于中间频谱，达到平均速度。基于两阶段的检测器，ThunderNet是最慢的。此外，下图显示了COCO数据集上所有检测头和主干（72种组合）的速度、准确性和参数权衡。

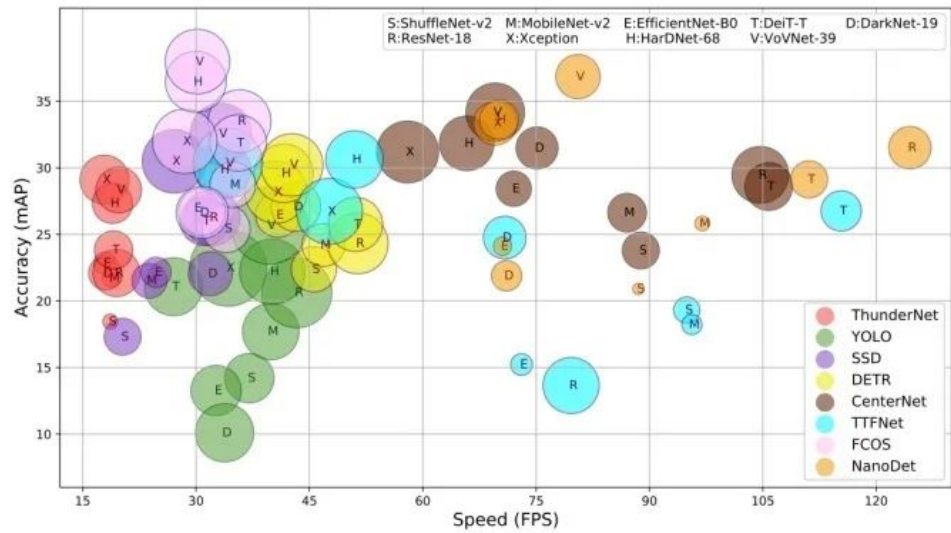


Figure 13: Speed-accuracy trade-off of all combinations of detection heads and backbones from Table 4 on COCO dataset. The size of the bubble corresponds to the number of parameters present in the model. Colors indicate detection heads and the letters represent different backbones.

每个气泡的大小表示网络中的参数量。大多数组合在15到50FPS的速度范围，而NanoDet和CenterNet为所有主干网实现了更高的速度。在骨干网中，低频谱网络消耗的能量最少因为这些网络的规模非常小，这也反映在较少的参数量上。DeiT-T证明是非常节能的。

在这些检测头中，除了FCOS之外，高频谱检测器的表现相当不错，因为FCOS比其他anchor-free的检测器更重型。NanoDet消耗的能量最少，因为其专门设计用于在移动硬件上运行。SSD和DETR在能耗方面保持中等频谱。ThunderNet除了分类和回归阶段外，还有proposal阶段，比单阶段检测器消耗更多的能量。详细分析：下表提供了两个不同数据集VOC和COCO的八个检测头、九个主干的更详细信息。

Table 4: Results of all detection models with various backbones on COCO and VOC datasets. MAC count (G), number of parameters (#P; in Million), inference energy consumption (in kilo Joules), accuracy in terms of F1 score and mAP, and inference speed in terms of FPS are reported. mAP is reported @IoU: [0.5 - 0.95] for COCO and @IoU: 0.5 for VOC. The best performance for each detector model is in bold and the overall best performance for each metric and dataset is highlighted.

Head	Backbone	COCO						VOC					
		MAC (G)	#P (M)	Inf Energy (kJ)	F1 Score	mAP	Inf Speed (FPS)	MAC (G)	#P (M)	Inf Energy (kJ)	F1 Score	mAP	Inf Speed (FPS)
ThunderNet	ShuffleNet-v2	1.71	2.51	32.31	36.52	18.45	18.65	1.41	2.21	12.69	71.26	68.24	41.41
	MobileNet-v2	2.71	4.12	33.83	40.66	21.73	18.87	2.41	3.81	16.23	74.85	72.14	41.92
	EfficientNet-B0	2.91	5.60	38.32	42.06	22.83	17.93	2.60	5.29	24.26	77.11	74.76	36.67
	DeiT-T	2.23	12.58	54.21	40.62	22.07	18.01	1.92	12.27	31.41	74.45	72.01	35.63
	DarkNet-19	15.57	24.83	37.93	41.15	22.09	19.47	15.26	24.52	22.81	78.75	76.59	44.91
	ResNet-18	17.54	16.33	37.94	43.56	23.83	19.09	17.24	16.02	24.20	77.49	75.12	44.41
	Xception	25.48	26.78	61.56	49.89	29.11	17.88	25.17	26.47	36.76	81.64	79.92	34.05
	HarDNet-68	23.32	18.04	42.19	47.75	27.32	18.95	23.01	17.73	27.51	77.70	75.28	40.18
	VoVNet-39	38.16	23.11	53.66	48.99	28.34	19.77	37.85	22.80	30.62	81.00	79.31	41.21
YOLO	ShuffleNet-v2	7.44	27.22	15.65	34.33	14.20	37.05	7.36	26.91	7.97	36.91	51.04	90.60
	MobileNet-v2	10.24	35.70	15.65	40.16	17.71	39.88	10.16	35.39	12.22	71.45	71.19	56.75
	EfficientNet-B0	8.28	28.23	18.76	32.58	13.27	32.65	8.20	27.92	15.78	41.25	57.26	75.66
	DeiT-T	8.08	37.85	30.52	27.06	10.10	33.78	8.00	37.54	28.42	64.35	63.35	43.86
	DarkNet-19	22.53	54.63	17.96	44.07	20.62	43.33	22.33	50.66	18.07	71.14	73.76	71.12
	ResNet-18	43.03	37.58	25.21	45.43	21.07	27.02	23.25	41.23	21.15	69.82	69.55	58.24
	Xception	34.75	65.73	37.38	47.74	22.49	34.26	34.67	65.43	30.20	73.26	75.88	43.80
	HarDNet-68	30.26	47.69	24.40	46.90	22.20	40.09	30.18	47.39	25.52	75.04	76.86	48.70
	VoVNet-39	54.79	66.78	33.31	51.02	25.68	39.67	44.99	52.43	24.67	76.03	78.61	53.94
SSD	ShuffleNet-v2	4.24	15.14	24.16	37.45	17.31	20.28	1.92	8.21	14.55	72.35	67.38	22.40
	MobileNet-v2	4.30	13.60	26.22	43.97	21.52	23.84	2.51	6.03	18.08	78.56	72.54	48.53
	EfficientNet-B0	3.52	10.16	26.68	31.77	22.15	24.72	2.46	6.46	19.86	79.54	73.17	45.92
	DeiT-T	5.41	21.53	35.39	48.98	22.05	31.89	2.51	14.41	21.83	79.31	74.54	52.92
	DarkNet-19	21.44	35.63	31.54	50.72	26.27	32.06	16.56	27.06	16.52	83.17	77.84	71.13
	ResNet-18	21.35	26.47	31.81	50.87	26.00	31.10	17.95	18.45	17.22	82.00	76.47	74.00
	Xception	33.55	44.93	47.56	55.59	30.49	27.08	27.04	30.96	27.03	82.59	78.86	50.05
	HarDNet-68	30.85	35.49	35.34	55.79	29.86	33.53	24.91	24.99	21.07	85.37	80.41	60.12
	VoVNet-39	48.66	41.70	40.77	58.88	32.55	33.28	40.59	30.37	21.20	85.80	81.57	65.29
DETR	ShuffleNet-v2	0.99	22.98	16.93	47.27	22.38	45.58	0.99	22.97	11.90	77.52	69.72	46.97
	MobileNet-v2	1.85	20.75	18.88	49.81	24.20	46.85	1.85	20.73	14.91	78.15	70.78	48.24
	EfficientNet-B0	2.04	22.21	26.56	53.68	26.45	40.82	2.04	22.20	23.17	81.95	75.31	42.76
	DeiT-T	0.73	27.21	30.64	53.09	27.08	43.21	0.73	27.20	27.09	80.11	73.07	43.95
	DarkNet-19	14.60	41.29	25.09	49.72	24.32	51.40	14.60	41.28	21.59	81.00	74.29	51.58
	ResNet-18	16.57	32.79	26.32	51.68	25.75	51.10	16.57	32.77	21.32	79.68	72.68	50.88
	Xception	24.45	43.20	36.46	54.90	28.17	40.57	24.45	43.19	31.51	83.13	77.32	41.39
	HarDNet-68	22.50	38.41	30.86	56.44	29.59	41.61	22.50	38.40	27.57	83.70	77.44	41.64
	VoVNet-39	37.30	43.45	34.23	57.88	30.22	42.67	37.30	43.43	28.97	84.57	78.67	44.08
CenterNet	ShuffleNet-v2	12.04	15.20	19.31	48.20	23.79	88.85	11.92	15.19	16.48	76.36	68.42	79.71
	MobileNet-v2	12.90	16.71	20.84	52.21	26.63	87.04	12.77	16.70	15.70	80.11	72.84	89.44
	EfficientNet-B0	13.09	13.76	23.61	54.29	28.43	72.04	12.97	13.76	18.69	81.05	73.14	73.65
	DeiT-T	11.78	19.65	24.12	58.05	31.49	75.12	11.65	19.64	18.48	81.03	75.24	75.14
	DarkNet-19	25.65	36.08	19.35	55.62	29.47	104.67	25.53	36.07	15.32	82.66	76.47	103.19
	ResNet-18	27.63	25.22	19.05	54.49	28.59	105.78	27.50	25.21	14.69	81.68	75.18	104.98
	Xception	35.50	42.70	29.33	57.35	31.19	57.99	35.38	42.69	24.25	82.73	77.30	57.07
	HarDNet-68	33.55	33.20	25.97	58.19	31.84	65.82	33.42	33.19	20.81	82.69	77.49	65.70
	VoVNet-39	48.35	38.23	26.25	60.92	34.17	69.59	48.23	38.23	22.05	84.88	79.73	64.98
TFNet	ShuffleNet-v2	5.55	7.66	15.33	44.15	19.32	94.93	5.44	7.65	12.08	75.93	68.37	94.49
	MobileNet-v2	2.72	4.50	15.14	41.74	18.22	95.61	2.68	4.50	11.89	78.37	71.56	97.45
	EfficientNet-B0	3.12	5.26	19.17	37.71	15.22	73.07	3.08	5.26	15.90	65.32	60.97	74.23
	DeiT-T	10.91	20.35	24.85	50.37	24.76	70.85	10.86	20.34	19.52	79.67	74.41	70.57
	DarkNet-19	46.77	35.16	23.40	34.31	13.65	79.63	46.52	35.14	18.16	83.62	77.62	79.80
	ResNet-18	24.81	18.15	18.05	53.27	26.76	115.37	24.68	18.14	13.57	81.62	75.31	113.08
	Xception	59.15	48.72	34.43	54.73	26.73	47.65	58.89	48.70	29.49	81.43	74.70	46.75
	HarDNet-68	65.57	36.64	32.27	57.52	30.63	50.96	65.26	36.62	26.72	85.17	80.15	50.00
	VoVNet-39	160.17	53.98	46.65	57.77	30.36	34.20	159.66	53.94	42.60	86.05	81.20	33.11
FCOS	ShuffleNet-v2	106.58	25.55	43.45	50.73	25.43	33.99	105.07	25.27	37.98	79.95	71.41	33.67
	MobileNet-v2	107.24	23.28	42.96	54.71	28.72	34.85	105.73	23.00	37.27	83.56	76.96	34.88
	EfficientNet-B0	107.77	24.38	47.93	52.99	26.70	30.75	106.26	24.15	33.25	83.53	77.77	30.46
	DeiT-T	107.06	30.40	48.64	54.02	26.64	30.81	105.55	30.12	43.46	84.22	77.19	30.72
	DarkNet-19	120.23	44.09	43.05	59.82	33.50	35.76	118.73	43.81	37.33	86.12	79.61	35.78
	ResNet-18	122.34	35.51	43.80	58.32	31.87	35.59	120.83	35.24	37.36	84.85	77.98	35.58
	Xception	130.70	46.50	51.94	57.75	31.99	28.50	129.19	46.22	46.53	86.26	80.08	28.44
	HarDNet-68	128.76	41.43	49.16	63.67	36.45	29.96	127.25	41.16	43.56	87.86	81.59	30.03
	VoVNet-39	144.04	46.63	49.65	64.63	37.98	30.18	142.53	46.36	43.73	88.01	81.79	30.37
NanoDet	ShuffleNet-v2	1.04	1.43	10.66	45.66	20.90	88.54	1.00	1.41	7.28	75.88	66.81	87.45
	MobileNet-v2	1.89	2.45	13.79	52.09	25.82	96.99	1.86	2.44	10.80	81.92	74.67	98.91
	EfficientNet-B0	2.18	3.74	18.39	50.48	24.11	70.55	2.15	3.72	15.49	82.69	75.45	70.70
	DeiT-T	1.58	10.31	22.99	47.91	21.89	71.09	1.55	10.29	18.84	81.47	75.11	71.64
	DarkNet-19	14.76	20.09	14.45	59.27	31.51	124.46	14.72	20.07	11.55	84.32	79.13	128.48
	ResNet-18	16.83	15.32	16.41	55.80	29.12	111.12	16.79	15.30	13.21	82.87	76.15	112.33
	Xception	24.48	21.19	24.84	61.22	33.35	69.61	24.45	21.18	19.59	85.59	79.94	69.99
	HarDNet-68	22.64	16.83	21.21	61.00	33.61	70.09	22.61	16.83	21.21	85.25	79.77	71.43
	VoVNet-39	37.53	21.89	22.26	64.84	36.85	80.54	37.50	21.87	17.32	86.09	81.67	80.58

FCOS+VoVNet-39组合具有最高的准确度，而NanoDet+DarkNet-19组合具有最高的推理速度。Transformer组合DETR+DeiT-T有最低的MAC计数，因为DETR在单个特征图上工作，而最新的DeiT-T（带有5M参数）可谓足智多谋。两个数据集的资源占用是相似的，除了在SSD中，在将数据集从VOC更改为COCO时参数量增加（在某些情况下约为2倍）。由于SSD使用六个特征图，每个特征图都有单独的anchor，因此当类数量增加时会产生资源开销。这种效果在anchor-free设计中的放大程度较小。

总体而言，在主干网中，高频谱网络HardNet-68和VoVNet-39在所有指标上都表现良好，Xception除了精度外，在所有指标上都表现不佳。中间频谱由ResNet-18、DarkNet-19和DeiT-T组合

成，在精度、速度和资源占用之间取得了良好的平衡。DeiT-T因为没有卷积，是资源占用最友好且MAC数量最少的。

在检测头中，NanoDet实现了高精度和速度，同时还具有较高的计算效率。CenterNet和TTFNet也提供了良好的平衡，而TTFNet有更快的训练时间。DETR（同样，NanoDet）在与较轻的主干配对时显示较低的MAC计数。为了进一步证明在不同指标上评估网络的重要性，正所见，精度与所有其他指标正相关，GMAC在F1得分之后是最高相关的。速度仅与推理能耗高度负相关。能耗与GMAC的正相关性最高，表明MAC操作较多的网络往往会消耗更多的能量。

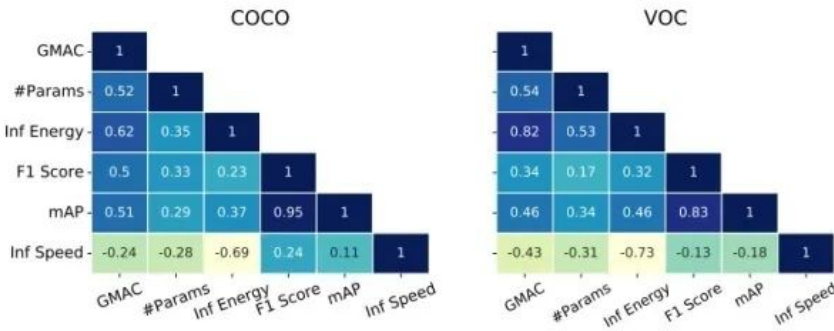


Figure 14: Correlation between different parameters reported in Table 4 for all heads and backbones.

9 解耦影响

目标尺寸的影响

对于大多数目标检测器来说，小目标的检测是一个具有挑战性的问题（Liu et al., 2021）。为了展示网络在不同尺度目标上的性能，作者比较了三种不同大小（即小、中、大）的主干和检测头的精度。下图显示所有主干和检测头的组合在不同尺度的上表现。TTFNet、NanoDet和FCOS优于其他网络，主要是因为得到性能最佳的重型主干网的加持，如HardDNet-68或VoVNet-39。重型主干的更高分辨率的特征图与这些检测头中的FPN/PAN相结合能得到更好性能，使得中型和大型目标的精度要好得多。在检测头中，FCOS和NanoDet对各种尺寸的目标总体表现更好。TTFNet、CenterNet和SSD，位于中频谱，配上更快的主干网络对于需要更高推理速度的应用来说是个不错的选择。FCOS的稳定性能归功于其更重型的架构，其使用了五个不同尺度的特征图。为了进一步分析，作者考虑所有具有HardNet-68主干的检测头，因为它在更复杂的数据集COCO上提供了最佳平衡。

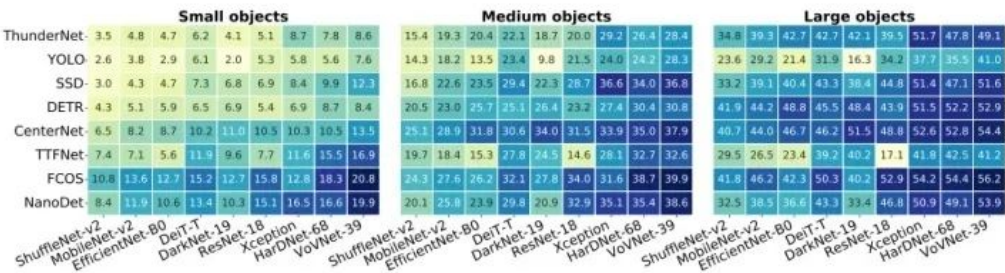


Figure 15: Accuracy (mAP) of all detection networks in conjunction with various backbones reported on different object sizes (small, medium and large) @IoU: [0.5-0.95] on COCO dataset. Darker colors represent higher accuracy values.

输入图像尺寸的影响

用于训练的输入图像的分辨率对最终精度起着重要作用。所有先前实验中使用的图像分辨率均为12×512。为了分析其他输入分辨率在精度和速度上的权衡，作者使用不同图像尺寸进行训练，包括256、384、512和736。图像尺寸选择为“16的偶数倍”。

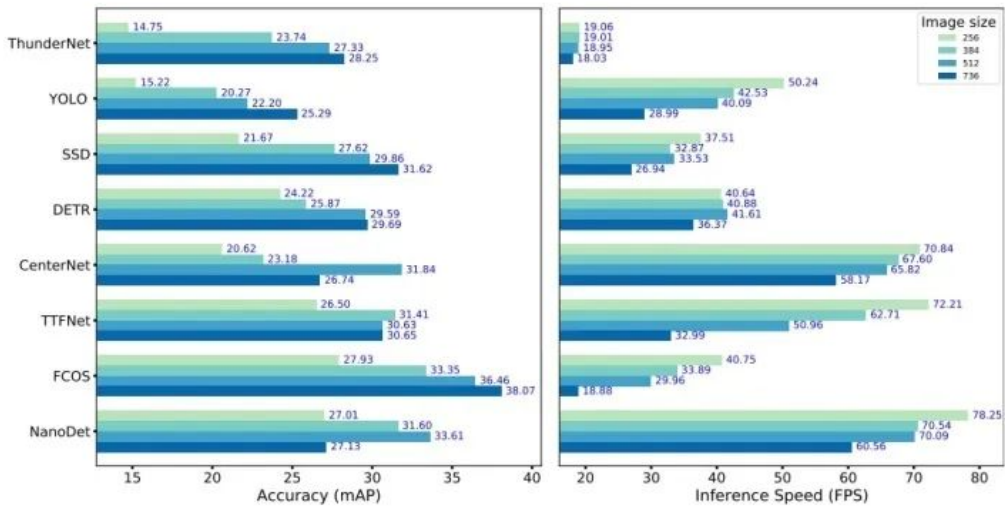


Figure 16: Effect of image sizes on accuracy and inference speed of detection models with HardNet-68 backbone on COCO dataset. As image size increases, the gain in accuracy is masked by the decline in speed.

上图表明，检测头的精度遵循“收益递减”的趋势。在大多数情况下，从256到384的图像分辨率有显著的精度跳跃。但是，随着图像尺寸的进一步增加，增益会降低，当图像尺寸从512变为736时增益最小（在某些情况下，它也会降低精度）。此外，作者观察到更高图像尺寸的精度增益被更大的速度下降所掩盖。例如，FCOS中从512到736的4.4%精度增益被37%的速度降低所掩盖，因此FCOS切换到高分辨率不是最优选择。对于YOLO、TTFNet和FCOS，速度随着分辨率的增加而明显下降，因为YOLO和其它FPN中的多分辨率尺度特征连接中的操作数量随着图像大小的增加而增加。DETR使用注意力块来捕获图像的全局上下文信息，而ThunderNet具有单独的区域每个样本的proposal，它们对不同的图像大小不太敏感。

anchor尺寸的影响

anchor大小和纵横比需要与数据集中存在的目标的大小一致，因此是anchor-based网络中的重要参数。anchor大小也需要先验，因此很难适应新的数据集。在作者研究的八个检测头中，SSD、YOLO和ThunderNet使用anchor-based的方法进行检测。为了分析anchors对检测性能的影响，作者对所有三个anchor的检测头进行了实验，这些检测头具有不同的anchor尺寸，由它们各自的宽度和高度定义。目的是为anchor的宽度和高度添加一些偏移量，并分析网络速度和准确性的变化。作者不是线性增加/减少anchor尺寸，而是从具有不同sigma的高斯分布中采样偏移量，并将其添加到原始anchor的宽度和高度以创建修改后的anchor尺寸。此外，修改后的anchor在整个特定实验中保持不变。原始anchor的宽度和高度（来自这些网络的原始架构）被视为基线。

下表显示了修改后的anchor相对于基线（第一行）的精度和推理速度方面的变化。作者观察到不同大小的anchor的精度变化遵循随机模式，精度和速度之间没有相关性。ThunderNet在精度上，对anchor框大小的变化不敏感，而其推理速度不断提高。SSD对这些变化非常敏感，因为所有三组anchor框大小的准确性都降低了，而其中一个的速度增加了。这些变化提高了YOLO的精度，但并不是都影响其推理速度。anchor尺寸影响检测的非确定性方式证明，修改anchors以提高检测效果不是一项简单而直接的做法。

Table 5: Effect of anchor box sizes on accuracy (mAP) and inference speed (FPS) for the three anchor-based detection networks with HardNet-68 backbone on COCO dataset (@IoU: [0.5, 0.95]). Numbers are reported in terms of percentage change w.r.t the original anchor sizes. The best performance for each detector is in bold. There is no common pattern and the performance changes in a non-deterministic way with respect to the anchor box sizes.

Anchor size	ThunderNet		YOLO		SSD	
	mAP	FPS	mAP	FPS	mAP	FPS
Orig.	27.32	18.65	22.20	40.09	29.86	33.53
0.2 σ	+0.26	+4.00	+3.38	-2.51	-0.21	-2.46
0.3 σ	+1.06	+4.90	+2.32	-6.70	-4.29	+7.86
0.5 σ	+1.79	+4.72	+3.69	+2.38	-9.76	-2.09

置信度阈值的影响

目标检测器会产生许多框，并使用一个阈值来过滤掉冗余和低置信度预测。改变这个阈值会影响准确率和召回率。因此，置信度阈值在计算精度和推理速度方面起着至关重要的作用。由于未明确提及此类参数，因此从先前的目标检测文献中再现结果存在差异。使用不同的阈值显示出精度和推理数量的显著差异。ThunderNet有基于区域的proposal，并利用Soft-NMS，分数衰减而不是固定硬阈值，因此这个参数不影响结果。CenterNet和TTFNet使用maxpool来选择预测，而DETR去除了传统的检测模块，因此不使用这个阈值。因此，本研究仅考虑了YOLO、SSD、FCOS和NanoDet。下表显示了使用较高阈值时准确度的下降以及使用较低阈值时速度的降低。例如，通过将阈值从0.01更改为0.4，YOLO的mAP下降了~22%，而速度提高了~71%。

Table 6: Effect of confidence threshold on performance and inference speed (FPS) on detectors with HardNet-68 backbone on COCO dataset (@IoU: [0.5-0.95]). The results show that a lower threshold yields higher accuracy while a higher threshold yields higher speed.

Head	Threshold 0.01			Threshold 0.4		
	mAR	mAP	FPS	mAR	mAP	FPS
YOLO	53.74	22.20	40.09	30.51	17.15	68.52
SSD	66.26	29.86	33.53	39.05	24.32	49.95
FCOS	76.74	36.45	29.96	43.97	29.24	31.12
NanoDet	74.20	33.61	70.09	44.73	28.26	72.15

可变形卷积的影响

引入了可变形卷积(DCN)层，有助于检测具有几何变形的目标。传统的卷积根据定义的内核大小在图像上使用固定的矩形网格。在DCN中，每个网格点都可以移动一个可学习的偏移量，即网格是可变形的。DCN基准测试主要关注精度的提升，而不是其他指标。为了获得有关速度和资源需求的更多信息，作者分析了DCN层对在其最初提出的架构中使用DCN的两个检测器的影响，即CenterNet和TTFNet。下表提供了上述两个检测器的精度、速度、参数数量和能耗并且没有DCN层。在两个数据集COCO和BDD上测试结果。在BDD数据集上，将CenterNet中的DCN层替换为标准卷积层，导致准确率下降1.8%，速度提升8%以上。DCN层的使用也增加了参数量，导致能耗增加13%。在COCO数据集上，将TTFNet中的DCN改为标准卷积层后，精度下降不到5%，而速度提高了10%，能耗提高了约6%。这些结果表明，使用DCN层时存在固有的精度、速度和资源需求权衡。

Table 7: Effect of DCN layers on accuracy and efficiency metrics. The detectors with HardNet-68 backbone are trained and tested on COCO and BDD datasets (@IoU: [0.5-0.95]). The gain in accuracy using DCN comes at the cost of lower speed and higher resource requirements and energy consumption.

Head	DCN Layer	COCO				BDD			
		mAP	Inf Speed (FPS)	#Params (M)	Energy (kJ)	mAP	Inf Speed (FPS)	#Params (M)	Energy (kJ)
CenterNet	✓	31.84	65.82	33.20	25.97	22.12	66.57	33.19	43.69
	✗	29.67	71.77	32.76	24.72	21.72	72.43	32.76	38.58
TTFNet	✓	30.63	50.96	36.64	32.27	22.01	55.83	36.22	51.81
	✗	29.13	56.10	36.16	30.34	20.98	56.98	36.14	51.13

10 目标检测器的可靠性

许多应用程序，尤其是对安全至关重要的应用程序，需要检测网络高度准确和可靠。检测器不仅必须精确，还应该指出它们何时可能不正确。模型校准提供了对模型不确定性的洞察，随后可以将其传达给最终用户或协助进一步处理模型输出。它是指与一个预测相关的概率反映整体精度可能性的度量。大多数工作只专注于提高网络的预测精度，但必须有一个经过良好校准的模型。大型且精度高的网络往往过于自信（Guoetal.,2017）并且校准错误。因此，迫切需要重新审视和测量SOTA检测器的校准，以获得完整的评估。校准的大部分工作都集中在分类领域，但Kuppers等人。（2020）包括边界框预测以及分类标签，以评估检测器的整体校准。预期校准误差(ECE)(Naeniatal.,2015)是衡量校准的常用指标之一，用于衡量预测置信度和准确度之间的期望差异。在分类领域，该分数表示分类准确度与估计的信心。检测ECE(D-ECE)(Kuppersetal.,2020)测量观察到的平均精度(AP)与分类和边界框属性的偏差。置信空间和边界框空间被划分为相等的bin，通过迭代所有bin并在每个bin中累积AP和置信度之间的差异来计算D-ECE。一维案例只考虑置信度，但作者使用多维D-ECE案例，它结合了所有因素：p、cx、cy、w、h，分别表示预测的类别概率、中心坐标、宽度和高度。

可靠性图(DeGroot&Fienberg,1983)用于直观地表示模型校准，其中准确度被绘制为置信度的函数。下表和图分别提供了可靠性分数和图表。

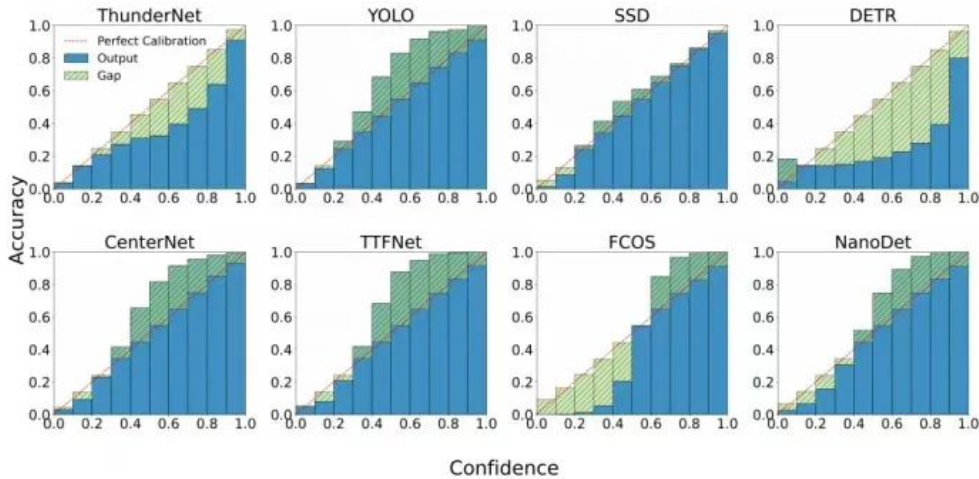


Figure 17: Reliability diagrams (based on classification prediction) of detection networks with HarDNet-68 backbone trained and tested on COCO dataset. Green shaded areas indicate the error compared to perfect calibration - darker shade of green indicates the underconfident predictions whereas the lighter shade indicates the overconfident predictions. The better the calibration, the more reliable the predictions are. SSD is relatively well calibrated. In general, the single-stage detectors are more underconfident while the two-stage ThunderNet and DETR are very overconfident.

Table 8: Calibration error for detection networks with HarDNet-68 backbone trained and tested on COCO dataset. Each row shows the error for a specific dimension: classification confidence only (ECE) and confidence + bounding box (D-ECE). The lowest calibration errors across all detectors are highlighted.

Head	ThunderNet	YOLO	SSD	DETR	CenterNet	TTFNet	FCOS	NanoDet
ECE (%)	8.09	5.88	4.07	18.46	4.31	6.17	21.56	6.48
D-ECE (%)	14.94	10.92	7.60	23.48	8.04	16.62	22.20	9.02

在可靠性图中，对角线表示完美校准，绿色阴影表示校准中的差距。在anchor-based的检测器中，SSD校准得很好，而YOLO则更不自信。所有基于关键点的方法（上图中的最后一行）都更倾向于不自信，并且对他们的预测更加谨慎，因此可能更适合安全关键型应用。但是，基于Transformer(DETR)和基于两阶段(ThunderNet)的检测器过于自信，在安全关键型应用中可能不受欢迎。当还包括定位时，校准误差会增加（如D-ECE中所反映的）。作者注意到有几个分类领域的校准解决方案，例如直方图分箱(Zadrozny&Elkan,2001)、逻辑校准/普拉特缩放(Plattetal.,1999)、温度缩放(Guoetal.,2017)和beta校准(Kulletal.,2017)。然而，将这些应用于目标检测可能没有那么有效，因此已经提出了其他工作（Neumann等人，2018年；Kuppers等人，2020年）来采纳专门针对目标检测的经过良好校准的估计。在这项研究中，作者专注于比较不同检测器的可靠性，而不是深入研究解决方案以改进其校准。

11 自然鲁棒性

自动驾驶等实时目标检测应用非常重视安全性和精度。在此类应用中使用的目标检测器需要在其预测中保持一致，并且对各种因素（例如不断变化的天气条件、光照和各种其他成像效果）具有鲁棒性。公共数据集没有充分覆盖所有这些影响，因此作者通过在它们上添加不同的损坏来模拟它们。Corrupted COCO数据集创建有15种不同的损坏。下图显示了每个检测头在四种损坏类别上的结果：噪声、模糊、天气和数字化影响。精度值是该特定类别中不同损坏的平均值。这些 level0是网络在原始数据上的表现。所有网络的性能在所有损坏上都会恶化，并且随着严重性的增加而下降得更快。在噪声、模糊和数字化影响方面，与天气类别相比，这些网络的性能下降幅度相对较大。对于所有损坏类别，FCOS是最稳健的，而YOLO是最不稳健的。就IID数据的准确度而言，检测器的顶部、中部和低谱在OOD设置上仍然保持良好。FCOS在IID测试集中被证明是最准确的，即使在具有挑战性的OOD设置（即自然损坏的数据）上也能保持这种性能。

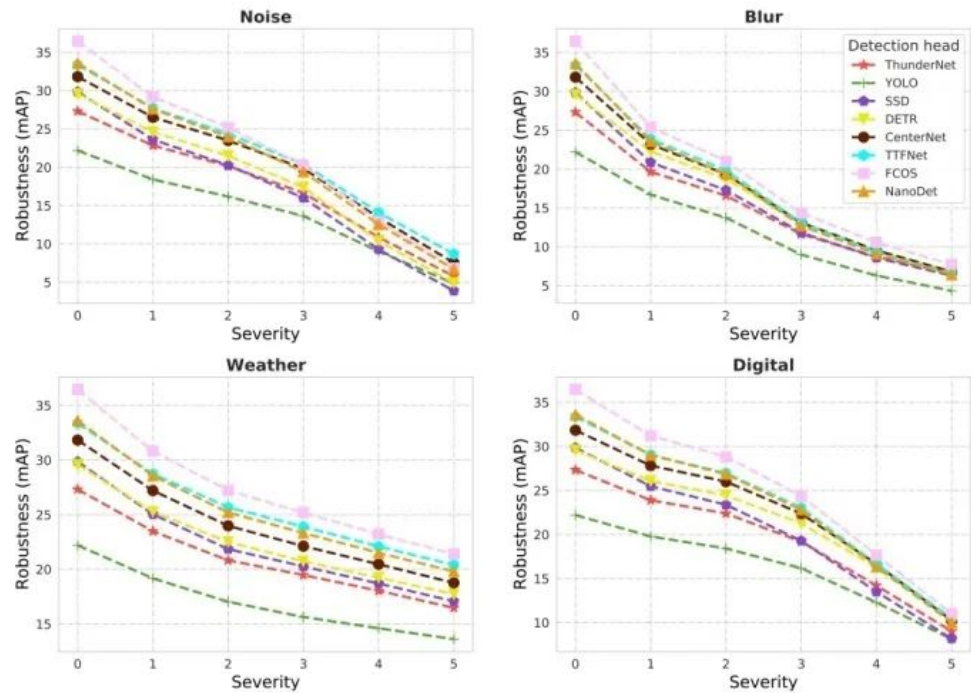
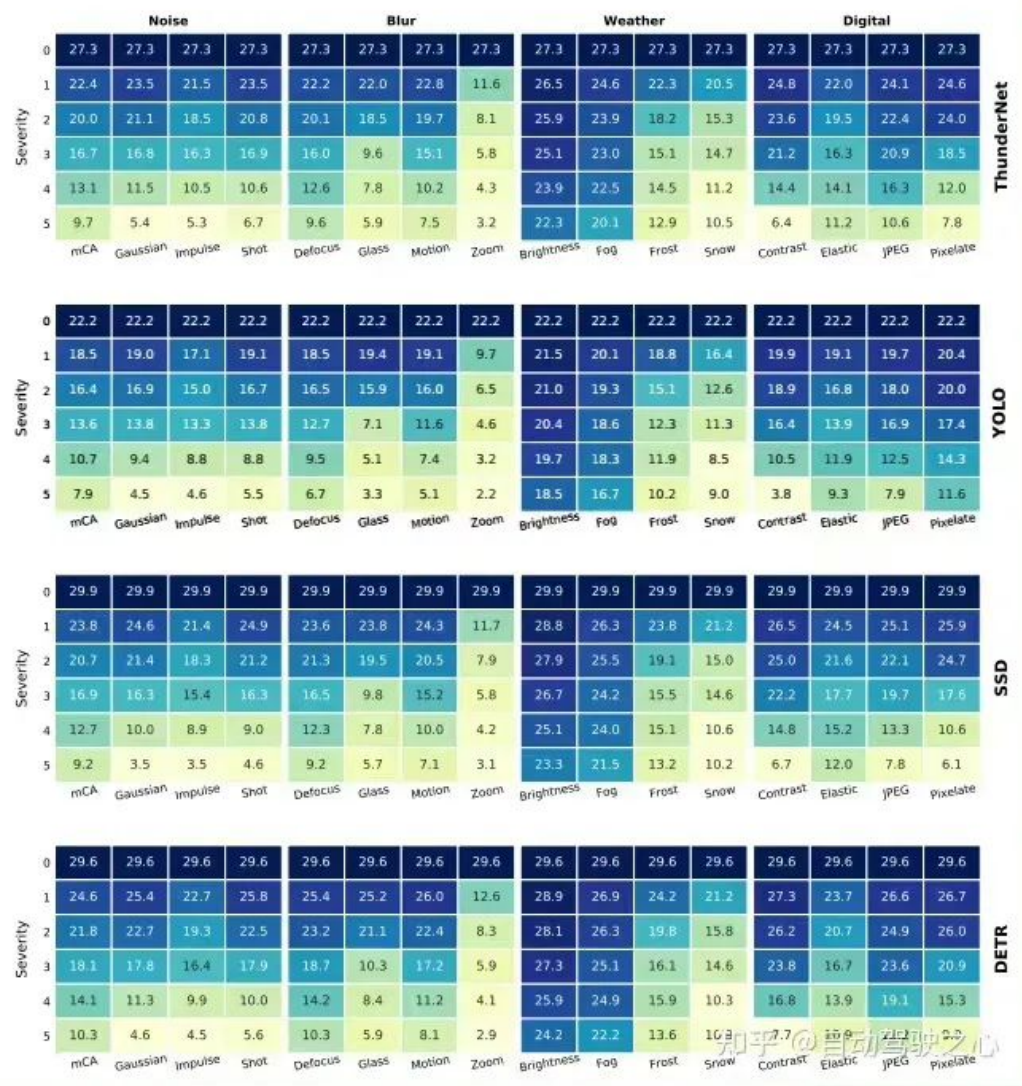


Figure 18: Robustness of detection networks with HardNet-68 backbone trained on COCO and tested on Corrupted COCO dataset. Results are shown on four categories of corruptions: Noise, Blur, Weather, and Digital.

为了提供更详细的分析，作者在下图中显示了每个网络的所有15种不同损坏的结果。在每个热图中，作者通过对所有损坏进行平均来计算平均损坏准确度(mCA)。所有检测器在所有三种噪声（高斯噪声、散粒噪声和脉冲噪声）上都显示出类似的性能下降趋势。与其他噪声相比，FCOS和TTFNet的下降最少，并且对噪声损坏相对更稳健。在模糊损坏中，散焦和运动模糊的下降更为稳定，而对于玻璃模糊，精度最初逐渐下降，但在严重级别3之后急剧下降。在变焦模糊中，所有检测器的性能下降都从严重级别1开始。与霜和雪相比，所有检测器对不同亮度和雾的破坏都具有鲁棒性。最差的性能出现在下雪的条件下，并且趋势相似。在数字效果中，与像素化和对比度相比，网络对弹性变换和JPEG压缩的鲁棒性更强。所有模型对对比变化的鲁棒性都较低，而YOLO是最不鲁棒的。



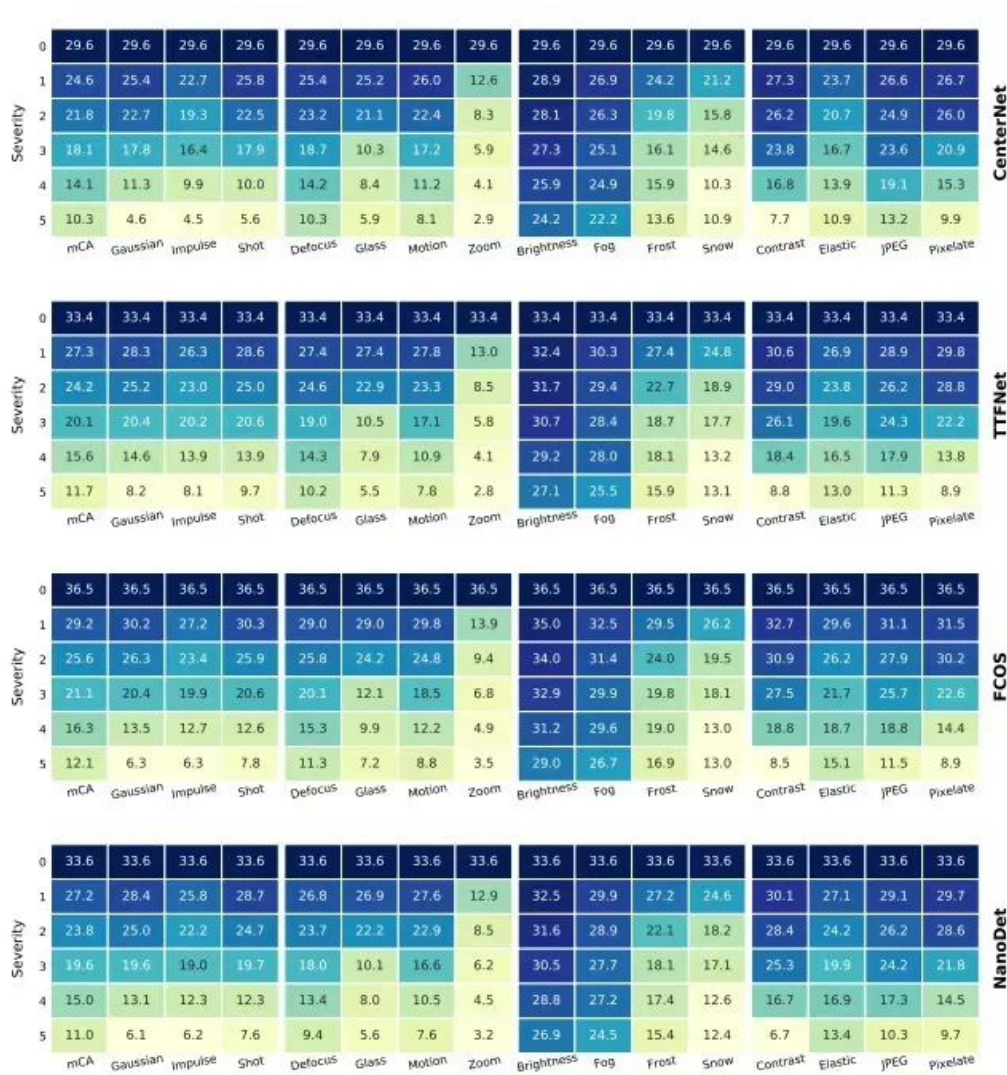


Figure 22: Robustness accuracy of all the eight detection heads with HardNet-68 backbone when tested on Corrupted COCO dataset. The corruption has 5 levels of severity, with level 0 being the result on original COCO data.

12 对抗鲁棒性

一些工作已经表明深度神经网络对抗性攻击的脆弱性。对抗性扰动是难以察觉的噪声，当添加到数据中时，人眼无法察觉，但可能导致网络做出错误的预测。在自动驾驶等安全关键型应用中，稳健性对于防止网络做出不合时宜的决策更为重要。因此，对抗鲁棒性是目标检测的关键指标。然而，它在文献中并不突出。在这里，作者评估了所有八个检测器网络对抗性攻击的鲁棒性。

作者采用基于梯度的攻击，利用网络的梯度信息来产生扰动。投影梯度下降(PGD)(Madry et al., 2017)是一种常见的非目标攻击，它最大化训练损失以产生对抗性扰动，该扰动被限制在epsilon范围内。作者同时使用分类损失和回归损失作为PGD攻击的目标。作者以不同的攻击强度执行PGD攻击，并在下图中展示精度。Epsilon=0时的精度是指原始测试集上的干净准确度。随着攻击强度的增加，性能下降。与其他检测器相比，CenterNet和DETR表现出稳定且更好的鲁棒性。FCOS具有最高的自然精度，并且对非常弱的攻击表现出良好的抵抗力，但在更高的扰动下性能急剧下降。TTFNet和ThunderNet表现次之。YOLO、NanoDet和SSD占据下一个频谱。

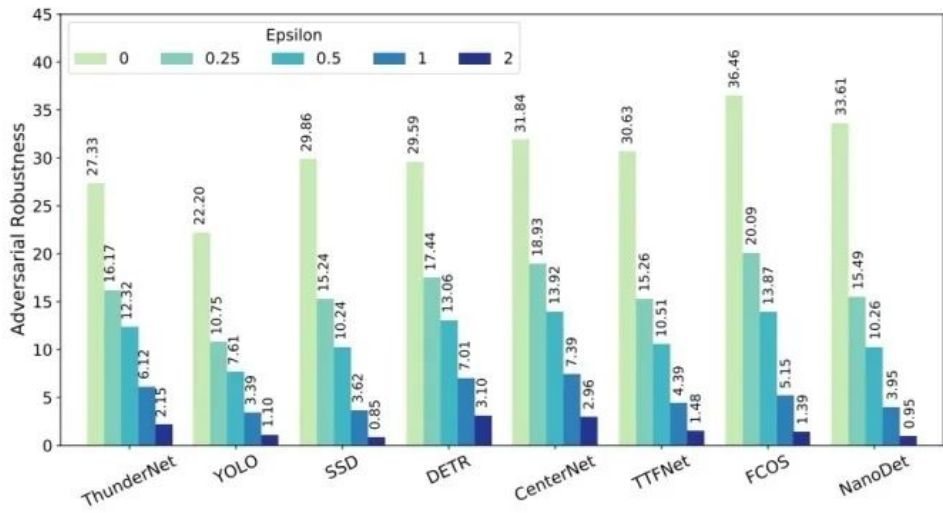


Figure 19: Adversarial robustness of all detection networks (with HardNet-68 backbone) trained on COCO, against PGD attacks of varying strengths (Epsilon). Epsilon=0 represents the original natural accuracy.

13 案例研究：自动驾驶

实时目标检测在自动驾驶(AD)领域具有高度相关性，网络需要学习各种目标，例如城市道路和高速公路上的行人、车辆和路标。检测网络的大多数基准都在VOC和COCO上提供数据集，主要由家常目标组成。这些数据集的结果不足以衡量网络在AD场景中的性能。因此，作者使用BDD数据集(Yuetal.,2018)对AD进行了实际案例研究，该数据集是该领域中最大和最多样化的数据集之一。

首先，作者展示了这个复杂数据集上所有网络的性能。然后，作者通过使用在BDD上训练的模型并在不同的数据集（即Cityscapes(Cordtsetal.,2016)）上进行测试来解决分布外(OOD)泛化问题。最后，作者将所有在BDD上训练的模型部署在嵌入式设备上，并展示每个网络的实时应用能力。因为与速度下降相比，DCN获得的准确度并不显着。因此，在本节中，作者统一考虑所有没有DCN层的网络。

下表是作者展示了在BDD验证集上获得的结果。类似于赵等人。(2018a)，作者以IoU=0.7计算了模型的准确度(mAP)。NanoDet表现出最好的准确度。FCOS是次之最准确的，但速度慢，能耗也最高。CenterNet速度最快，但准确度略低。SSD消耗的能量最少，YOLO有准确率最低。有趣的是，BDD数据集中目标位置的偏差导致生成的区域建议更少，从而使ThunderNet更快。

Table 9: Results of detection models with HardNet-68 backbone trained on BDD and tested on BDD (in-distribution test) and CityScape (out-of-distribution test) datasets. MAC count, number of parameters, inference energy consumption (kJ), mAP (@IoU: 0.7), and inference speed (FPS) are reported. The best performance for each metric is highlighted.

Head	MAC (G)	#Params (M)	BDD			Cityscapes		
			mAP	FPS	kJ	mAP	FPS	kJ
ThunderNet	22.96	17.68	21.51	45.35	47.94	18.69	44.26	4.44
YOLO	30.17	47.34	6.38	42.01	50.53	4.47	36.46	4.44
SSD	24.27	23.46	26.15	10.69	29.97	20.07	11.58	3.16
DETR	22.50	38.40	13.80	40.10	60.13	11.29	36.26	5.45
CenterNet	36.58	32.76	24.19	73.18	44.23	19.22	65.62	4.86
TTFNet	69.57	36.14	23.05	58.56	52.94	18.74	50.09	5.69
FCOS	126.99	41.11	27.89	30.63	93.23	23.95	28.68	9.16
NanoDet	22.60	16.81	30.09	55.51	47.12	22.88	48.55	4.69

分布转移的泛化是AD场景中的主要挑战之一。网络在实际应用中部署时，需要适应看不见的数据并始终如一地执行。然而，大多数深度学习基准都显示在测试集上，其分布与训练数据相同（Geirhosetal.,2020）。因此，为了测试网络对分布变化的鲁棒性，作者在Cityscapes数据上测试了BDD训练模型。作者从Cityscapes数据集的实例分割注释中提取真实边界框。作者观察到FCOS的准确度最高，NanoDet紧随其后。CenterNet是最快的网络，SSD在这两个集合中是最节能的。一般来说，anchor-free的检测器是跨具有挑战性的AD数据集泛化的较好选择。

AD应用程序具有功率和资源限制，因为网络部署在板载边缘设备上。检测网络在低功耗设备上的实时性能对其功效至关重要。对于部署，作者使用TensorRT库将网络转换为优化的高性能推理引擎。TensorRT是NVIDIA的并行编程模型，可以优化神经网络以部署在嵌入式或汽车产品平台上。然后，这些引擎在NVIDIA的三个不同范围的GPU上进行测试：(1)2080Ti，一种常用的桌面GPU，(2)Jetson-Xavier，一种强大的移动GPU，以及(3)Jetson-TX2，一种低功耗的移动GPU。下表显示了所有8个检测器在三种精度模式下的推理速度，即FP32、FP16和INT8。

Table 10: Inference speed (FPS) of all TensorRT optimized detection networks with HardNet-68 backbone on BDD dataset deployed on three devices at FP32, FP16, and INT8 precision (note that INT8 is not supported on TX2). The best performances are highlighted.

Head	RTX 2080Ti			Xavier			TX2	
	FP32	FP16	INT8	FP32	FP16	INT8	FP32	FP16
ThunderNet	151	212	225	15	30	36	10	16
YOLO	174	244	280	15	33	45	9	17
SSD	154	212	225	15	30	36	10	16
DETR	138	184	184	14	29	30	8	13
CenterNet	141	229	228	11	29	29	7	13
TTFNET	101	193	206	7	19	19	4	8
FCOS	53	116	133	4	10	13	2	4
NanoDet	153	210	213	15	27	33	9	14

性能趋势可能与之前看到的不同，因为它取决于TensorRT对不同层的优化。优化融合了后续层并使计算并行化。anchor-based的检测器ThunderNet、YOLO和SSD具有相对简单的架构，优化后的速度增益最高。YOLO是最简单的，得到的优化最多，是所有平台上最快的。然而，所有anchor-free的检测器从优化中获得的速度增益最小。DETR位于中间频谱，并且由于transformer架构相对较新，它不像其他卷积层那样被TensorRT引擎优化。这个独特的案例研究表明，性能在一台设备上看到的趋势不一定会转化为其他硬件。该基准在选择模型以部署在边缘设备上以实现实时AD应用程序时非常有用。

14 案例研究：健康领域

深度学习的最新进展使人工智能模型能够帮助外科医生和放射科医生诊断和治疗危及生命的疾病。手动检测需要专业知识，需要时间，并且可以也会受到人为错误的影响。基于AI的检测解决方案有助于降低成本和资源，并可以为医学成像中的检测提供准确的工具。其中一个应用是使用DNN检测医学图像中的息肉。结肠和直肠（结肠直肠）癌通常是由结肠或直肠内层的息肉引起的。

检测这些息肉并在早期阶段对其进行治疗对于癌症治疗至关重要。医学图像的分布与COCO和VOC等标准数据集截然不同。因此，标准基准可能无法提供有关为此应用程序选择哪种模型的重要信息。此外，不同的指标更相关，具体取决于应用程序。虽然标准基准侧重于准确性的精度指标，但在医疗保健行业，即使是一个假阴性也可能比假阳性结果造成更大的损害，召回更为重要。为了解决这种新的数据分布和指标，作者专门针对通过评估Kvasir-SEG数据集上的检测器来评估医学图像。下表显示了在Kvasir-SEG的testsplit上获得的结果。召回与此应用程序更相关，因此，作者将平均平均召回(mAR)与mAP一起报告。某些网络（如YOLO）可能没有最高的精度，但召回低。FCOS具有最高的召回率和精度，这使其成为此类测试用例的理想候选者。在速度方面，SSD是最快的，而Nanodet次之。

Table 11: Results of detection networks with HardNet-68 backbone trained and tested on Kvasir-SEG. MAC count, number of parameters, inference energy consumption, mAR and mAP (@IoU: [0.5-0.95]), and inference speed (FPS) are reported. The best performance for each metric is highlighted.

Head	MAC (G)	#Params (M)	Inf Energy (kJ)	Inf Speed (FPS)	mAR	mAP
ThunderNet	22.92	17.63	4.34±0.02	51.79±2.01	89.53±0.25	67.49±1.14
YOLO	30.16	47.29	5.05±0.02	53.16±0.95	94.10±1.02	60.25±1.34
SSD	23.02	21.66	4.02±0.02	68.70±1.27	90.42±0.25	66.98±0.82
DETR	22.50	38.39	5.07±0.05	37.85±1.11	89.97±0.68	63.70±1.04
CenterNet	33.38	33.19	4.44±0.04	55.99±1.02	92.33±0.68	69.30±0.93
TTFNet	65.16	36.62	5.31±0.06	43.58±0.47	91.30±0.51	65.58±1.30
FCOS	126.78	41.07	6.76±0.03	26.76±0.54	94.21±0.28	71.42±0.25
NanoDet	22.60	16.80	4.07±0.03	65.02±1.20	91.59±0.00	70.44±1.08

15 讨论

作者在跨不同数据集的统一实验设置下对特征提取器和检测器的组合（范围从两阶段、单阶段、anchor-based、anchor-free到基于Transformer的架构）进行了全面研究。作者得出了一组广泛的结果，包括精度、速度、资源和能耗，以及稳健性和校准分析。作者评估了检测器对两种自然对抗性破坏的鲁棒性。此外，还突出显示了详细的见解，以全面了解不同变量对最终结果的影响。对不同的变量，如主干网络的影响、图像大小、目标大小、置信阈值和特定架构层进行了解耦和研究。作者还就两个不同的行业贡献了两个独特的案例研究：自动驾驶和医疗保健。


作者进一步在嵌入式硬件上优化和基准测试网络，以检查网络部署在边缘设备上的可行性。结果表明，anchor-free的检测器倾向于很好地泛化多个数据集，因为不再需要对anchor进行优化。NanoDet在准确性和速度方面都很好，同时对资源也很友好。CenterNet是第二快的，并且在所有其他指标上也处于良好的范围内，TTFNet位于中间范围内。FCOS的准确性最高，但在其他指标上表现不佳，而DETR是基于Transformer的检测器具有最低的MAC计数，位于中间频谱。在主干网中，专门设计的现代网络对于低内存流量，例如HardDNet，在精度、推理速度和能耗之间提供最佳平衡。所有检测器在检测小目标时都表现不佳，FCOS的表现相对更好。不同的anchor以非确定性的方式影响性能，因此难以泛化。

作者指出在切换到更高图像尺寸或使用DCN层时应考虑的精度-速度-资源要求权衡。在对抗自然损坏的鲁棒性上，所有网络的性能在所有15次损坏上都下降了，并且随着严重性的增加下降得更快。一般来说，anchor-free的检测器比其他检测器对自然损坏的鲁棒性相对更强。FCOS是最鲁棒的，而YOLO是最不鲁棒的。FCOS和TTFNet对嘈杂和模糊的损坏相对更鲁棒，但所有检测器在雪天条件下表现都很差。CenterNet被证明是最强大的对对抗性扰动具有鲁棒性，而FCOS和DETR对这些攻击也具有很强的抵抗力。在可靠性分析方面，SSD的校准相对最佳，而anchor-free的检测器在预测中更加谨慎，因此使其在安全关键型应用中比较推荐。ThunderNet和DETR倾向于更加过度自信。

作者对基于深度学习的实时目标检测网络在不同数据集和不同域上全面的分析。广泛的分析了新架构（Transformer vs. CNN）的能力和缺陷。不同的应用有不同的标准，作者的研究可以作为工业界衡量不同标准的指南，在为各自的应用选择检测器时可以进行权衡。而且由于新的检测网络正在不断的出现，作者也希望能启发研究人员将这项研究作为设计新网络的参考准则。本研究强调了标准化、透明和公平的重要性，同时强调需要将重点从名义上的改进转移到更开阔的视野。

参考

[1] A Comprehensive Study of Real-Time Object Detection Networks Across Multiple Domains: A Survey



极市

EXTREME MART

视觉AI工程项目

实训周

—从开发到落地提升CV工程能力—

🕒


2022.9.24-10.12

形式

线上直播

+

社群全程实战技术指导



扫码即刻报名
提升CV工程能力

费用

0元

面向人群

有一定编程和算法基础的学生、在职人士

平台提供

免费算力，已标注真实数据集，产业端落地项目，在线技术答疑支持

参加视觉AI工程项目实训周将收获什么？

◆ 模型训练 > 模型、业务测试 > 封装SDK全流程技术能力提升

◆ 目标检测和图像分割方向真实落地的工程项目经验

◆ 10+实训项目任选一项，完成实训任务将获得 ¥ 200-3000 现金奖励

EXTREME MART

极市干货

算法竞赛：国际赛事证书，220G数据集开放下载！ACCV2022国际细粒度图像分析挑战赛开赛！

技术综述：浅聊对比学习（Contrastive Learning） | 深度学习图像分类任务中那些不得不看的11个tricks总结

极视角动态：极视角与华为联合发布基于昇腾AI的「AICE赋能行业解决方案」 | 算法误报怎么办？自训练工具使得算法迭代效率提升50%！



CV技术社群邀请函



△长按添加极市小助手

添加极市小助手微信（ID：cvmart2）

备注：姓名-学校/公司-研究方向-城市（如：小极-北大-目标检测-深圳）

即可申请加入极市 [目标检测/图像分割/工业检测/人脸/医学影像/3D/SLAM/自动驾驶/超分辨率/姿态估计/ReID/GAN/图像增强/OCR/视频理解](#) 等技术交流群

极市&深大CV技术交流群已创建，欢迎深大校友加入，在群内自由交流学术心得，分享学术讯息，共建良好的技术交流氛围。

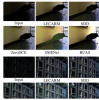


[点击阅读原文进入CV社区](#)
[收获更多技术干货](#)

阅读原文

喜欢此内容的人还喜欢

ICCV23 | 将隐式神经表征用于低光增强，北大张健团队提出NeRCo
极市平台



ICCV 2023 | 南开程明明团队提出适用于SR任务的新颖注意力机制（已开源）
极市平台



YOLOv5帮助母猪产仔？南京农业大学研发母猪产仔检测模型并部署到Jetson Nano开发板
极市平台

