

南京理工大学泰州科技学院

毕业设计说明书(论文)

作者: 王端虹 学号: 1909030229

学院(系): 计算机科学与工程学院

专业: 计算机科学与技术

题目: 基于注意力机制的
目标检测算法研究

指导者: 姜枫 教授

评阅者: 贾波 教授

2023 年 5 月

毕业设计说明书（论文）中文摘要

目标检测是计算机视觉领域重要的研究课题，其任务是检测出图像中所包含的物体，标定其位置并对物体进行分类。本文首先对近年来目标检测的方法进行研究和总结，分析各种方法的优缺点。本文以端到端的 DETR（DEtection TRansformer）模型为基础，针对其网络收敛速度慢和对图像中小物体检测效果不佳的弊端，结合 ResNet 中多尺度特征，借鉴可变卷积的思想，引入多尺度可变注意力模型，用以替换 DETR 中编码器的自注意力和解码器中的交叉注意力，提出多尺度可变 DETR，有效地解决了 DETR 的不足。通过与主流的目标检测方法进行实验对比，本文所提出的多尺度可变 DETR 的收敛速度快了十倍，检测效果也有了较为明显的提升。

关键词 目标检测 可变注意力机制 多尺度特征 Transformer

毕业设计说明书（论文）外文摘要

Title Research on Object Detection Algorithm Based on Attention Mechanism

Abstract

Object detection is an important research topic in the field of computer vision, whose task is to detect the objects contained in the image, mark their positions, and classify the objects. This thesis first studies and summarizes the methods of object detection in recent years, analyzing the advantages and disadvantages of various methods. This thesis is based on the end-to-end DETR (Detection TRansformer) model. In order to address the drawbacks of slow network convergence speed and poor detection performance for small objects in images, combined with the multi-scale features in ResNet and drawing inspiration from the idea of variable convolution, a multi-scale deformable attention model is introduced to replace the self attention in the encoder and cross attention in the decoder of DETR. A multi-scale deformable DETR (MSD DETR) is proposed, which effectively solves the shortcomings of DETR. Through experimental comparison with state-of-the-art object detection methods, the convergence speed of the multi-scale variable DETR is 10 times faster than original DETR, and the detection effect has also been significantly improved.

Keywords Object Detection Deformable Attention Multi-Scale Features
Transformer

目 录

1 绪论	1
1.1 研究背景	1
1.2 研究意义	3
1.3 本文工作	3
1.4 本文组织结构	4
2 相关工作	5
2.1 注意力机制	5
2.2 目标检测中的多尺度特征	6
2.3 Transformer 的多头注意力机制	6
2.3 ResNet	7
2.4 Faster R-CNN	8
2.5 YOLO	9
2.6 DETR	10
2.7 PyTorch	11
3 多尺度可变 DETR	13
3.1 模型结构	13
3.2 特征提取模块	13
3.3 可变注意力模型	14
3.4 多尺度可变注意力模型	16
3.5 可变编码器	18
3.6 可变换码器	18
3.7 预测层	19
4 实验及讨论	20
4.1 数据集	20
4.2 实验配置	21
4.3 实验指标	22
4.4 实验结果	22
4.5 可视化实验	25
5 结论与展望	27
5.1 结论与分析	27
5.2 展望	28
结束语	30
致 谢	31
参 考 文 献	32

1 绪论

计算机视觉（CV，Computer Vision）是当前计算机领域、人工智能领域一个热门的研究方向，根据任务的不同，计算机视觉的研究包含图像分类（image classification）、目标检测（object detection）、图像分割（image segmentation）等不同方向。其中，目标检测的任务是从图像中预测出所包含的目标，标定其位置区域并预测其所属类别。因此，相较于其他任务而言，目标检测更加注重对图片整体信息的获取、理解和分析，目标检测任务一般分为两个阶段，即目标定位和目标分类。

1.1 研究背景

目标检测是计算机视觉中一个重要的分支，其主要任务是检测出图片中的感兴趣的物体，标出物体的位置对物体进行分类。目标检测是实例分割（instance segmentation）、目标跟踪（target tracking）、语义分割（semantic segmentation）等任务的前序流程。

传统的目标检测方法通常包括三个步骤：第一步，区域选择；第二步，特征提取；第三步，目标分类。

区域选择是为了找出图像中目标所在的位置。最初人们采用滑动窗口（sliding windows，以下简称为滑窗）的方法对图像进行遍历，搜索图像中的感兴趣区域（region of intresting）。然而，该方法通过穷举的方式进行遍历，即便能够查找到所有目标会出现的位置，也会因为滑窗过多而产生时间复杂度高、冗余信息过多等问题，甚至会严重影响到了后续的特征提取和分类任务的速度。在实际应用中，通常采用几种固定的长宽比、大小的滑窗，因此，在实际检测中这种方法并不能完美地搜索出所有候选区域。

特征提取是从图像中候选区域提取特征并且转换为一种表征形式，即对图像候选区域进行编码。在图像领域，设计良好的特征应具备平移不变性（translation invariance）、旋转不变性（rotation invariance）和尺度不变性（scale invariance）等特性，以便于提高后续分类的准确率。因此，手工设计的特征（handcrafted features）比较难以达到以上的要求。

目标分类是对图像个检测出的目标区域进行分类。常见的分类器包括最近邻分类器（k-nearest neighbor）^[1,2]、逻辑回归（logistic regression）^[3,4]、支持向量机（support vector machines）^[5,6]、决策树（decision tree）^[7,8]、随机森林（random forest）^[9,10]等。

综上所述，传统的目标检测方法分为区域选择、特征提取和目标分类 3 个阶段，需要解决的难点有两个：第一，基于滑窗的区域选择方法具有很强的不确定性，无法保证能找到特定的目标，且窗口冗余、算法时间复杂度高。第二，手工设计的特征适应性不高，无法应对不同图片中不同目标的多样性变化。

近年来，深度学习技术被广泛应用于目标检测算法，该类方法可细分为两类：一种是基于回归方法的单阶段算法，另一种是基于候选框的两阶段算法。

两阶段目标检测算法包括 R-CNN、SPPNet、Fast R-CNN、Faster R-CNN、FPN 等。Girshick 等人^[11]提出 R-CNN 算法，首先使用 selective search^[12]从图像提取候选区域，随后将区域缩放到相同大小后使用在 ImageNet 训练好的 CNN 提取特征，最后使用支持向量机（SVM）分类。为了改善 R-CNN 在高密度候选区域反复提取特征，He 等人^[13]提出 SPPNet，在卷积层和全连接层之间添加空间金字塔池化（SPP，Spatial Pyramid Pooling），实现任意大小和长宽比的区域特征提取，并提升了检测速度。结合 R-CNN 和 SPPNet，Girshick 等人^[14]提出 Fast R-CNN，实现了在网络微调的同时对目标分类和边界框回归的同步训练。Ren 等人^[15]在 Fast R-CNN 的基础上，加入了区域生成网络（RPN，Region Proposal Network），实现了一个真正意义端到端、准实时的目标检测算法。Lin 等人^[16]结合顶层和中间层特征，提出特征金字塔（FPN，Feature Pyramid Networks），使算法在检测图像中不同尺度目标的性能大幅提升。

单阶段目标检测算法包括 YOLO 系列、SSD 等。Redmon 等人^[17]提出了一体化的卷积网络检测算法 YOLO，首先将任意大小图像调整为固定大小并进行网格划分，随后在每个网格预测物体的存在概率和可能位置，该网络需一次前向传播即可获得目标边界框位置，实现了实时检测。在此基础上，Redmon 等人^[18]通过增加 batch normalization、引入 Faster R-CNN 中的锚（anchor）机制、在 COCO 和 VOC 训练集上使用 k-means 聚类得到有先验的锚框形状、多尺度训练等方式改善了 YOLO 泛化能力弱和难以检测小物体的不足。YOLO v3^[19]进一步改进，使用逻辑回归对边界框置信度进行回归，用二分类交叉熵损失作为类别损失来处理多标签任务，进一步改善算法的性能和效率。Liu 等人^[20]提出 SSD（Single Shot MultiBox Detector）算法，吸取 YOLO 速度快和 RPN 定位准确的特点，在多分辨率特征图采用不同尺度和长宽比的先验框进行目标的实时检测，能够在检测精度和实时性上达到平衡。

随着深度学习的飞速发展，目标检测也抓住了这次的发展潮流从而焕发新生，基于 Transformer^[21]的 DETR 就是一个很好的例子。

DETR (DEtection TRansformer) [22]并没有像之前的 Faster R-CNN 或者 YOLO 一样将目标检测分为区域选择和分类两个任务,而是将目标检测视为一个集合预测问题。由于 Transformer 本质是将一个序列转换为另一个序列的模型,因此可以将 DETR 看作是一个从图像序列转换到一个集合序列的转换过程,该集合实际上就是一个可学习的位置编码。

如前所述,上述方法虽然都拥有着不错的准确率,并且在实时推理任务的时候也有一定的成绩,但这些模型都基于服务器设备设计所以需要大量的计算资源。因此,这些方法难以部署在边缘设备上运行。过去,人们尝试过采用高效组件和压缩技术来提高深度学习模型的检测效率。近期,一些学者提出轻量化的网络结构用于目标检测,这些网络旨在资源受限的情况下能够保证模型高效运行,如: mobileNet^[23]、MnasNet^[24]、OFA^[25]等。

1.2 研究意义

目标检测是计算机视觉领域的一个重要研究课题,是计算机视觉处理需要解决的最基本的问题之一。目标检测是很多下游任务的前置条件,它被广泛应用于人脸识别、工业检测、姿态监测等诸多领域。目标检测的精确度在很大程度上影响了它的下游任务的完成难度和完成质量,因此目标检测算法的研究对于计算机视觉领域来说具有重要的意义。近年来,目标检测一直是计算机视觉热门的研究方向,虽然全世界大量学者投入该课题的研究,但目标检测仍然面临着以下几个难点:

- (1) 图像中相同类别物体呈现不同形状的变化。
- (2) 图像采样时其质量受到光影变化和噪声的影响。
- (3) 图像中目标物体种类多,且各物体之间有重叠部分。
- (4) 特定场景下需要实时检测出物体,这点视频检测中尤为明显。

上述问题都在不同程度上影响了目标检测的准确性,所以,为了解决这些问题,近年来,越来越多的学者加入到目标检测的研究之中。

1.3 本文工作

2020 年 DETR 的提出给目标检测提供了一种新的研究思路,该方法将 Transformer 架构引入到目标检测问题,提出一种端到端的网络架构,并且取得了与 Faster R-CNN 不相上下的性能。然而,DETR 存在 2 个主要问题,分别是训练速度过慢和对局部特征不敏感。

本文以江苏省高校自然科学研究面向项目“面向砂岩图像的图像全景分割技术研究”（编号：19KJB520038）和江苏省高等学校大学生创新创业训练计划立项项目“基于 Transformer 的图像识别方法的研究”（编号：202213842037Y）为依托，对目标检测问题进行研究，在 DETR 的基础上进行优化，提出多尺度可变 DETR（Multi-scale Deformable DETR, MSD DETR），其主要创新点如下：

第一，在提取图像特征时提取了图像多层特征，参考了可变卷积（deformable convolution）的思路，设计了可变注意力机制匹配多尺度特征图，使用多尺度可变注意力替代了编码器的自注意力和解码器的交叉注意力。

第二，为了区分不同特征层的特征位置，设计了相对应的位置编码方式，引入参考点，在编码器增加参考点作为解码器的输入以实现可变注意力。通过算法的改进，在动物数据集上进行训练和测试，验证该算法的有效性。

1.4 本文组织结构

本文第一章为绪论，简述目标检测的定义、意义和国内外相关研究现状；第二章为相关工作，介绍了与课题相关的注意力机制、多尺度特征以及近段时间目标检测的典型算法；第三章提出模型 MSD DETR，详细阐述了模型整体结构、可变注意力机制原理的实现以及模型中每个模块的设计；第四章为实验结果及分析；第五章对全文进行总结并展望下一步研究方向。

2 相关工作

本章将介绍与本文方法相关的注意力机制、多头注意力机制、多尺度特征，并回顾近年来目标检测的经典算法 YOLO、Faster R-CNN 和 DETR 等。

2.1 注意力机制

注意力机制源于对人类视觉的研究。在认知科学中，人类只会选择性地关注所有信息的一部分，而忽略其他可见的信息，这种机制被称为注意力机制。这种机制被有效地应用到自然语言处理^[21]和计算机视觉中^[31]。

注意力机制分为自注意力（self attention）以及交叉注意力（cross attention）机制。在 Transformer 中，目前存在的问题是大量关键元素的高时间和内存复杂性^[26]，这在很多情况下阻碍了模型的可扩展性。最近，已经出现了三类解决这个问题的方法。

第一类是在关键元素上使用预定义的稀疏注意力，最直接的例子是将注意力模式限制为固定的大小。这是一种常见的后处理方法，将注意力模型限制在局部可以降低算法复杂度，但同时会丢失全局信息。为了补充全局信息，Child 等人^[27]引入了关键要素，以固定的时间间隔在关键元素中增加视野。Beltagy 等人^[28]允许少数特殊令牌访问所有关键元素。

第二类是学习数据依赖的稀疏注意力。Kitaev 等人^[29]提出了一种基于局部敏感哈希（LSH）的注意力，它将查询元素和关键元素散列在不同的容器中。Roy 等人^[30]提出了类似的想法，通过 k-means 找出最相关的键值。

第三类是探索子注意力中的低秩（low-rank）属性。Wang 等人^[30]通过在尺度而不是通道维度的线性投影来减少关键元素的数量。

在计算机视觉领域，高效注意力机制的设计主要局限于第一类。尽管在理论上降低了算法复杂度，但由于内存访问模式的固有限制，这种方法的实现速度比具有相同 FLOPS 的传统卷积慢得多。另一方面，正如 Zhu 等人^[32]所指出，一些卷积的变体，例如可变卷积^[33]和动态卷积^[34]，也可以看作是自注意力机制。特别地，可变卷积在图像识别方面比 Transformer 的自注意力机制更高效。与此同时，它还缺乏元素关系的建模机制。

基于上述原因，本文对注意力的改进方式属于第二类，只关注采样点周围的像素点，而不是计算全局元素。不同于 Ramachandran 等人^[36]对注意力机制的更改，在相同的 FLOPs 下，可变注意力仅比传统卷积慢一点点。

2.2 目标检测中的多尺度特征

目标检测的困难之一是如何有效地表示不同尺度的对象，目前，目标检测器通常利用多尺度特征来解决这一难题。FPN（Feature Pyramid Networks）^[36]作为开创性的作品之一，提出一种自上而下的路径来组合多尺度特征。PANet^[37]在 FPN 的顶部进一步添加了一条自下而上的路径。Kong 等人^[38]通过全局注意力操作结合了所有尺度的特征。赵等人^[39]提出了一个 U 形模块来融合多尺度特征。最近提出的 NAS-FPN^[40]和 AUTO-FPN^[41]通过神经架构搜索自动设计跨尺度连接。而本文提出的多尺度可变注意力机制可以通过注意力机制自动聚合多尺度特征图，无需借助特征金字塔即可适应多尺度特征图。

2.3 Transformer 的多头注意力机制

Transformer^[21]是一种基于机器翻译的网络结构，其模型结构如图 2.1 所示。

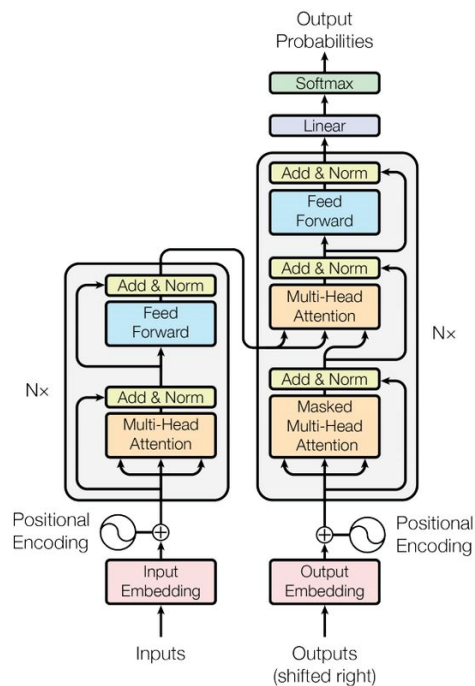


图 2.1 Transformer 模型结构图

对于一句话中的目标词和原词，Transformer 使用多头注意力衡量注意力权重聚合原词和目标词键对的兼容性。每个注意力头相互交互以达到信息共享的目的，这也是 Transformer 捕获全局信息能力强的原因之一。此处，目标词和原词分别对应模型结构中的 query 和 key。

给定一个查询元素 query 和一组 key 元素，多头注意力模块衡量注意力权重以自

适应的聚合 key 和 query 键对的兼容性。令 $q \in \Omega_q$ 表示 query 元素，其特征记为 $z_q \in \mathbb{R}^C$, $k \in \Omega_k$ 表示 key 元素，其特征记为 $x_k \in \mathbb{R}^C$ ，其中 C 是特征维度， Ω_q 以及 Ω_k 分别表示 query 和 key 元素的集合，多头注意力机制的公式为：

$$\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^M W_m [\sum_{k \in \Omega_k} A_{mqk} W'_m x_k] \quad (\text{公式2.1})$$

其中， m 是多头注意力的头数， $W'_m \in \mathbb{R}^{C_v \times C}$, $W_m \in \mathbb{R}^{C \times C_v}$ 均为可学习的参数，默认 $C_v = C/M$ 。注意力权重 $A_{mqk} \propto \exp\{\frac{z_q^T U_m^T V_m x_k}{\sqrt{C_v}}\}$ 归一化为 $\sum_{k \in \Omega_k} A_{mqk} = 1$ ，其中 U_m 和 $V_m \in \mathbb{R}^{C_v \times C}$ ，同样是可学习参数。

Transformer 的主要不足有两点：

第一，参数收敛速度慢。当 N_k 很大时权重 $A_{mqk} \approx \frac{1}{N_k}$ ，会导致输入特征的梯度不趋近于 0。因此 Transformer 往往需要很长的时间去训练，才能让注意力集中在指定的关键元素上。在图像领域，关键元素通常是图像的像素， N_k 可能非常大，收敛会变得非常慢。

第二，算法复杂度高。公式 2.1 的复杂度是 $O(N_q C^2 + N_k C^2 + N_q N_k C)$ 。在图像中，查询元素和关键元素都是像素， $N_q = N_k \gg C$ ，那么，多头注意力的复杂度可以记为 $O(N_q N_k C)$ ，也即多头注意力模型的复杂度会随着特征图大小的增长而增长。

2.3 ResNet

基于深度神经网络的机器学习算法被称作“深度学习”，随着深度学习的进一步发展，人们开始思考是不是层数越多，训练效果就越好。答案是否定的，如果网络只是简单的深度叠加，效果甚至会不如层次较低的网络。最初，解决该方法的方法是使用正则化初始化和 BN（Batch Normalization）层，然而，这样做的后果是会导致网络退化。

ResNet（残差网络）是一种深度卷积神经网络^[42]，其设计的关键是通过跨层连接（shortcut connection）来解决训练深度神经网络的梯度消失问题。网络的深度对模型的性能至关重要，当增加网络层数后，网络可以提取更加复杂的特征模式。然而当网络层数过深时，会出现网络退化的问题，准确率会出现饱和甚至下降的情况，网络的训练误差和测试误差均出现了明显的增长，而 ResNet 可以有效地解决这种退化问题。

ResNet 解决网络退化问题的原理如图 2.2 所示。图 2.2 表示 ResNet 中的残差模块，

残差模块分为两部分：主路径和跨层连接。图 2.2 左侧路径为主路径，通常由 2 或 3 个卷积层组成的，用于提取输入图像的特征信息。图 2.2 右侧路径为跨层连接,直接从输入连接到输出，即恒等映射。跨层连接的存在，使网络在计算残差反向传播时不会出现梯度消失或梯度爆炸等问题，从而提高了网络的性能和泛化能力。

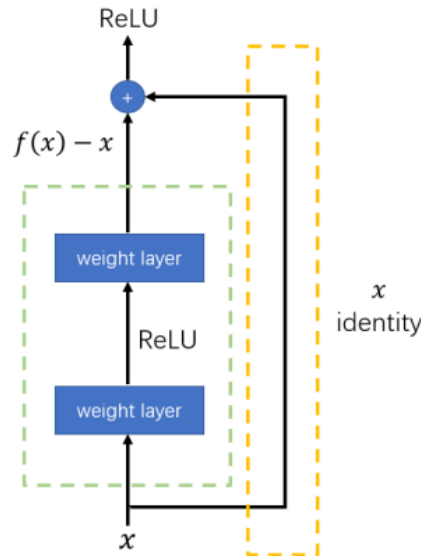


图 2.2 单个残差网络层

除此之外,ResNet通过在 3×3 卷积层前后引入 1×1 卷积层分别来减少和增加维度,让 3×3 卷积层成为瓶颈,减少参数和浮点运算,使模型更加轻量化。这样的改进使得网络深度突破100层,甚至诞生了100多层的极深网络,何凯明等人通过继承6个不同深度的ResNet(含两个ResNet-152),在ILSVRC 2015挑战赛上,其正确率高达96.43%,这已经超越了人的识别能力。

2.4 Faster R-CNN

在介绍Faster R-CNN之前先来了解一下其前身R-CNN和Fast R-CNN。

R-CNN^[43]打开了深度学习通往目标检测的大门,首先使用selective search^[12]算法从图像提取候选区域,随后将区域缩放到相同大小后使用在ImageNet训练好的CNN提取特征,最后使用支持向量机(SVM)分类。在Fast R-CNN^[14]中,仍然采用selective search方法产生候选区,并未使用到深度学习,并且是在CPU上进行训练的,非常耗时。其改进在于,在Fast R-CNN中一次性从图片中提取特征,并且把特征提取、分类、定位任务结合到一个网络进行训练,而R-CNN是分阶段完成。在Faster R-CNN^[15]中,为了降低产生候选区域的时间,彻底的放弃了selective search方法,使用RPN(Region Proposal Networks)生成候选区域,同时训练RPN和Fast R-CNN共享卷积

层，大幅提高了网络的检测速度。Faster R-CNN 网络结构图如图 2.3 所示。

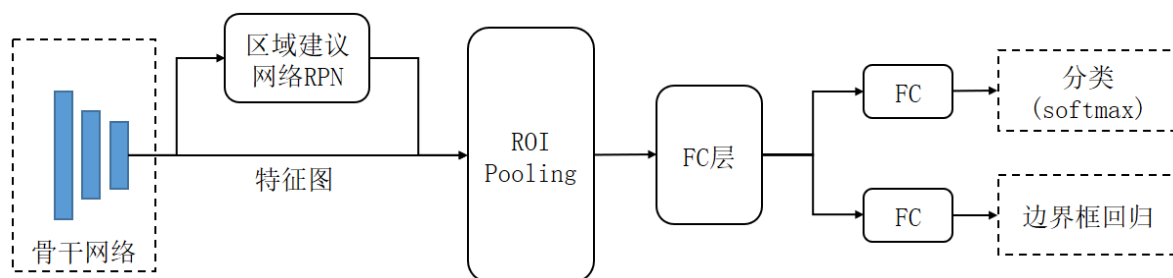


图 2.3 Faster R-CNN 网络结构图

从图 2.3 可以看出，Faster R-CNN 的工作流程如下：首先，输入一张图片，将其输入到一个共享卷积层进行特征提取；然后，用 RPN 生成 Anchor 框，对其进行剪裁过滤后通过 softmax 函数判断 anchors 是否是物体；随后，将特征图片送入一个特有卷积层再次提取特征，同时通过一个感兴趣区域池化层使每个感兴趣区域池化层生成固定尺寸的特征图；最后，通过一个全连接层得出分类和预测框。

2.5 YOLO

YOLO (You Only Look Once) 是单阶段、基于深度学习的回归方法的检测器，与 Fast R-CNN 相比，YOLO 在区域检测和分类流程上做到了统一。Faster R-CNN 需要反复的训练 RPN 网络和 Faster R-CNN 网络，也就是说它需要“看两眼”（提取候选框与分类分阶段训练）再确定结果，而 YOLO 将其统一为一个回归问题，也就是说 YOLO 只需要“看一眼”就可以得出候选框与分类了。

YOLO V1^[17]是在 2015 年由 Redmon 等人提出，首先将任意大小图像调整为固定大小并进行网格划分，随后在每个网格预测物体的存在概率和可能位置，该网络需一次前向传播即可获得目标边界框位置，实现了实时检测。但相比双阶段的 Faster R-CNN，其准确率和召回率略显不足。2017 年，YOLO 原作者提出了 YOLO V2，通过增加 batch normalization、引入 Faster R-CNN 中的锚 (anchor) 机制、在 COCO 和 VOC 训练集上使用 k-means 聚类得到有先验的锚框形状、多尺度训练等方式改善了 YOLO 泛化能力弱和难以检测小物体的不足。YOLO V2 比 YOLO V1 第一版更加准确，速度也更加快。随后作者提出了 YOLO V3，在 backbone 的选择上，选择了残差模型 Darknet-53，以及采用了 FPN 框架适应多尺度特征图，使用逻辑回归对边界框置信度进行回归，用二分类交叉熵损失作为类别损失来处理多标签任务。之后的 YOLO V4 在原有的框架上面做了大量优化，主要体现在数据处理、backbone、激活函数、损失函数等方面。而 YOLO V5 只是在 YOLO V4 的基础上进行修改优化，模型算法上并未

如图 2.5 所示，输入图像经过 CNN 网络后得到图像的特征矩阵；随后，将特征矩阵展平并添加位置编码；接着，输入编码器（transformer encoder）中学习特征的相关性信息；然后，将编码器的输出以及查询（object query）作为 decoder 的输入得到解码后的信息；再将解码后的信息传入 FFN（Feedforward Neural Network）得到预测信息；最后，判断 FFN 预测信息是否包含真实的目标对象，如果有，则输出预测框和类别，否则输出一个 no object 类。

DETR 简化了目标检测的流程，有效地消除了对人工设计组件的需求，如 NMS 或 anchor 生成。DETR 是一种基于集合的全局损失，通过二分匹配强制进行一对一预测，以及一种编码器-解码器架构大大优化了网络结构，可以很容易地推广到以统一的方式输出全景分割。DETR 在具有挑战性的 COCO 目标检测数据集上展示了与成熟且高度优化的 Faster R-CNN 基线相当的准确性和运行时间。

然而，DETR 也存在一定的问题，主要为：

（1）网络训练周期长。论文^[22]中指出，相较于 Faster R-CNN，DETR 的训练速度慢 10-20 倍。

（2）对小目标检测性能差。DETR 善于检测图像中大尺寸目标，如果用多尺度特征来解小目标，则大大提高 DETR 的时间复杂度。

2.7 PyTorch

PyTorch 是由 Facebook 是基于 Torch 开发的一个开源的用于神经网络框架。与 tensorflow 与 caffe 相比，Pytorch 支持动态神经网络，可以随意改变网络的行为，并且实现速度更快，同时还具有灵活性、易用性等优良特性。

PyTorch 的主要特点如下：

（1）动态计算图机制。PyTorch 的核心优势在于其动态计算图机制，该机制是指在 PyTorch 中，每个计算步骤都被定义为一个计算图节点，这些节点会组成一个单独的计算图，表示 Tensor 的计算和运算。这个计算图中节点的顺序和运算方式可以随时改变，并且可以由用户编写的代码动态控制。这使得动态计算图更加灵活，可以轻松处理控制流、递归等问题。

（2）灵活性强。在 PyTorch 的计算图中，节点可以由 Python 控制和修改，这意味着您可以轻松地在代码中创建条件语句、循环等逻辑结构，构建动态的计算流程。这种灵活性在 PyTorch 的应用中得到了极为广泛的认可，因为它能够更好地模拟真实世界中的问题和情况。

（3）易用性好。PyTorch 非常注重用户易用性，对代码进行了大量的优化和封装，使得它非常易于上手，并且有详细的文档和社区支持。此外，PyTorch 还提供了丰富的 API，包括各种优化器、学习率调度器、预训练模型等等，这些功能都极大地简化了编程过程，并减少了用户的工作量。

（4）性能优秀。PyTorch 在性能方面也有很大的优势，由于动态计算图的机制，它可以更好地利用现代计算机硬件（如 GPU、TPU 等）的计算能力，从而更快地计算大规模问题。此外，PyTorch 还针对多线程训练做了优化，进一步提高了训练效率。

3 多尺度可变 DETR

针对 DETR 存在的训练周期长、对小目标检测效果不佳的不足，本文提出多尺度可变 DETR（Multi-Scale Deformable DETR，简称 MSD DETR）对其进行针对性的改进。本章首先介绍 MSD DETR 的网络结构，然后分别介绍特征提取模块、特征编码器和预测模块，重点介绍可变注意力模块的原理和计算方法。

3.1 模型结构

本文所提出的 MSD DETR 模型结构如图 3.1 所示。该模型由 4 个部分组成，分别为特征提取模块、编码器、解码器和预测层。特征提取模块主要是 backbone（骨干网络），使用卷积神经网络用于预先从图像中抽取各尺度特征图。编码器中使用本文提出的多尺度可变注意力。解码器由多尺度可变注意力和自注意力组成。最后将解码层的输出作为预测层的输入，并在最后给出预测结果，同样是目标类别+预测框或者无目标两种输出。

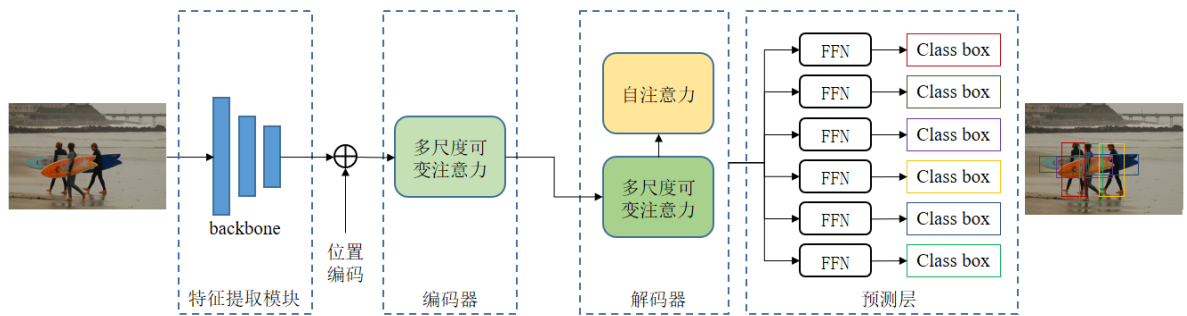


图 3.1 MSD DETR 网络架构图

3.2 特征提取模块

在 MSD DETR 中，特征提取模块网络的底层结构，主要用于提取数据的表面特征，也即通常所提的 backbone（骨干网络），通常采用主流的预训练的网络结构，常见的 backbone 有 VGG、Inception 和 ResNet 等。在本文中，选取在 ImageNet 21K 预训练的 ResNet 作为 backbone，主要基于两点考虑：第一，使用预训练的 backbone 可以缩短模型训练的时间；第二，使用预训练的 backbone 比重头训练一个模型的特征提取效果更佳优秀。

本文中，使用 ResNet 50 模型作为特征提取器。ResNet 50 包含 5 个大层：Conv1，Conv2，Conv3，Conv4 和 Conv5。本文使用 Conv3、Conv4 和 Conv5 层输出的特征图，分别记为 C3、C4 和 C5。此外，将 C5 经过一个大小为 3*3，步长为 2 的卷积得到特

征图 C6。至此，获得 C3、C4、C5、C6 四张特征图，然后将这四张特征图拼接到一起作为编码器层的输入。输入图像经过 backbone 以后，得到的 C3、C4 和 C5 层的输出特征图如图 3.2 所示。

表 3.1 ResNet50 模型

层名	特征图大小	参数
Conv1	112×112	[7×7,64], 步长为 2 3×3 池化, 步长为 2
Conv2	56×56	$\begin{bmatrix} 1\times 1, & 64 \\ 3\times 3, & 64 \\ 1\times 1, & 256 \end{bmatrix} \times 3$
Conv3	28×28	$\begin{bmatrix} 1\times 1, & 128 \\ 3\times 3, & 128 \\ 1\times 1, & 512 \end{bmatrix} \times 4$
Conv4	14×14	$\begin{bmatrix} 1\times 1, & 256 \\ 3\times 3, & 256 \\ 1\times 1, & 1024 \end{bmatrix} \times 6$
Conv5	7×7	$\begin{bmatrix} 1\times 1, & 512 \\ 3\times 3, & 512 \\ 1\times 1, & 2048 \end{bmatrix} \times 3$

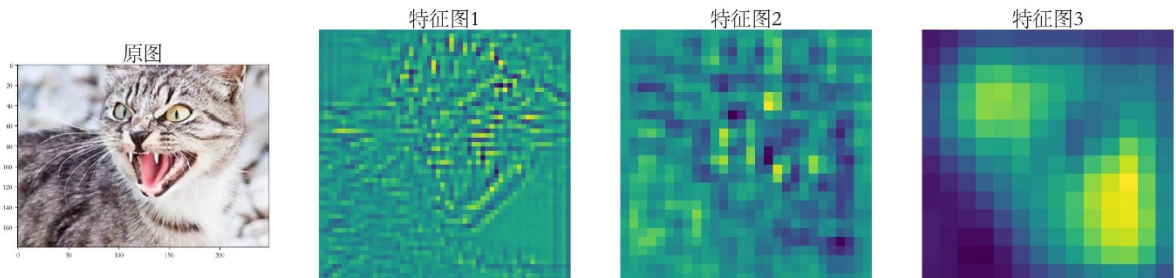


图 3.2 Backbone 特征图样例

3.3 可变注意力模型

如上所述，Transformer 中的注意力模型关注特征图中所有位置，即需计算所有图像块之间的注意力权重，是一种全局注意力机制。事实上，对每个图像块而言，与之有重要相关性的图像块只是图像中的一部分，而非全部，因此全局注意力机制既不符合实际情况，更是一种十分耗时的运算。

为了解决这个问题，受可变卷积网络 (Deformable Convolutional Networks, DCN)^[44] 的启发，本文提出可变注意力模型，其模型结构如图 3.3 所示。可变注意力模型的核心是它不关注所有可能的空间位置，仅关注每个参考点 (reference point) 周围一小部分采样点 (sampling points)。对于图像中的每个查询 query，为其分配少量固定数

量的 key 元素，可以一定程度上解决模型收敛（时间复杂度）和空间分辨率（空间复杂度）的问题。

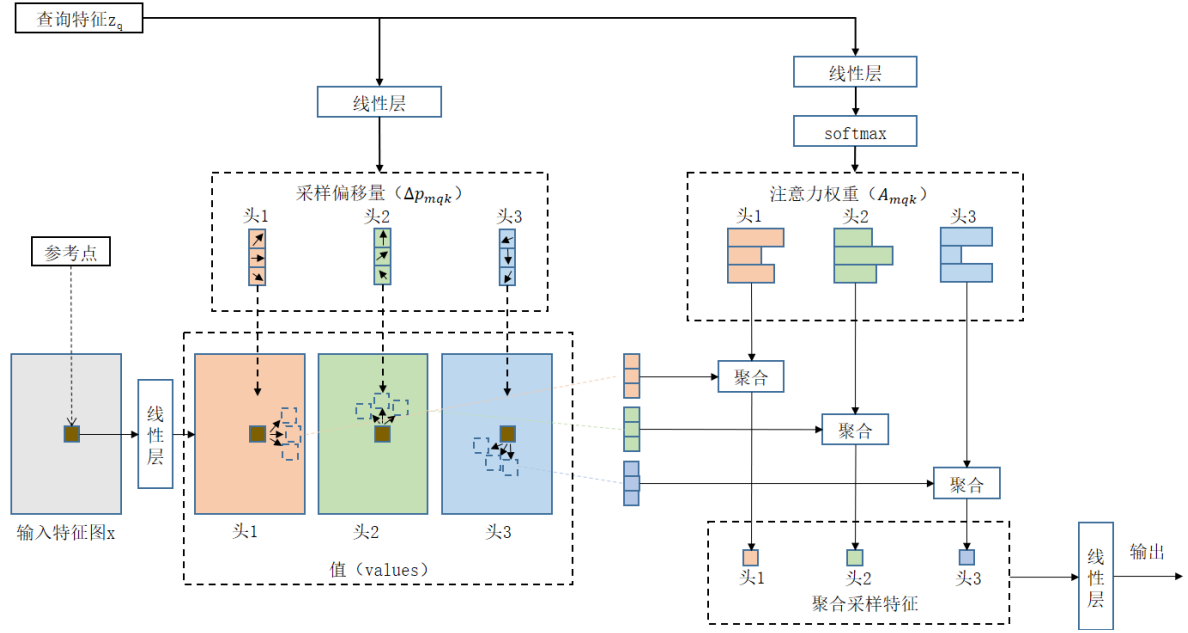


图 3.3 可变注意力模型结构图

对于给定的特征图 $x \in \mathbb{R}^{C \times H \times W}$ ，令 z_q 为查询元素也就是提取的图像特征，其特征记为 $z_q \in \mathbb{R}^C$ ， p_q 为一个二维参考点，则可变注意力的计算公式为：

$$\text{DAttn}(z_q, p_q, x) = \sum_{m=1}^M W_m \left[\sum_{k=1}^K A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right] \quad (\text{公式 3.1})$$

其中， m 代表多头注意力的下标， M 代表多头注意力的头数， k 代表采样点的下标， K 是采样点的总数， Δp_{mqk} 代表第 m 个注意力头中的第 k 个采样点的样本偏移量， A_{mqk} 代表第 m 个注意力头中第 k 个采样点的注意力权重。 A_{mqk} 的取值范围是 $[0, 1]$ ，通过 $\sum_{k=1}^K A_{mqk} = 1$ 做归一化处理。 $\Delta p_{mqk} \in \mathbb{R}^2$ 是无范围限制的二维实数。如图 3.3 所示，查询特征 z_q 经过线性投影，其通道数为 $3MK$ ，其中前 $2MK$ 个通道用来作采样偏移量的编码，也即 Δp_{mqk} ；剩下的 MK 个通道就用来作为注意力权重 A_{mqk} 的输入，并经过一个 $\text{softmax}()$ 函数后获取。

令 N_q 为查询元素的数量，当 MK 相对较小时，通过分析公式 3.1，不难得出可变注意力的时间复杂度为 $O(2N_q C^2 + \min(HWC^2, N_q KC^2))$ 。在图像编码器中，显然 $N_q =$

HW ，则此时可变注意力的时间复杂度为 $O(HWC^2)$ ；在解码器中，可变注意力使用交叉注意力使用，此时 $N_q = N$ （ N 为最大可检测到的目标数量），时间复杂度就变为 $O(NKC^2)$ ，也即在解码器中时间复杂度与特征图大小 HW 无关。

3.4 多尺度可变注意力模型

如前所述，DETR 对图像中大目标的检测效果较好，而容易忽略图像中尺寸较小的目标。因此，为了弥补 DETR 的这一缺陷，本文在可变注意力的基础上加上多尺度特征，以帮助检测图像中较小的目标。目前，增加多尺度特征的方法大多是通过在框架中增加 FPN^[36]来处理。在本文中，ResNet 50 提取的特征分为 C3、C4、C5 和 C6，将之与可变注意力模型相结合，即可得到多尺度可变注意力（Mutli-Scale Deformable Attention, MSD Attention）。

令 $\{x^l\}_{l=1}^L$ 为输入的多尺度特征图，其中 $x^l \in \mathbb{R}^{C \times H_l \times W_l}$ 。令 $\hat{p}_q \in [0,1]^2$ 为每个查询元素 q 的参考点归一化坐标，则多尺度可变注意力模块计算公式为：

$$\text{MSDAttn}\left(z_q, \hat{p}_q, \{x^l\}_{l=1}^L\right) = \sum_{m=1}^M W_m \left[\sum_{l=1}^L \sum_{k=1}^K A_{mlqk} \cdot W'_m x^l \left(\Phi_l \left(\hat{p}_q \right) + \Delta p_{mlqk} \right) \right] \quad (\text{公式 3.2})$$

其中， m 代表多头注意力的下标， k 代表采样点的下标， l 为输入特征图层级的下标。 Δp_{mlqk} 代表第 l 层特征中第 m 个注意力头中的第 k 个采样点的样本偏移量， A_{mlqk} 代表第 l 层特征中第 m 个注意力头中第 k 个采样点的注意力权重。对 A_{mlqk} 做归一化处理，使之满足 $\sum_{l=1}^L \sum_{k=1}^K A_{mlqk} = 1$ 。为了将多尺度信息表述得更加清楚，将查询信息进行归一化处理，使得 $\hat{p}_q \in [0,1]^2$ ，此处(0,0)代表图像左上角坐标，(1,1)代表图像右下角坐标。函数 $\Phi(\hat{p}_q)$ 的作用是对第 l 层特征图中归一化的坐标 $\hat{p}_q$ 进行缩放。

观察公式 3.1 与公式 3.2 可以发现，多尺度可变注意力和单尺度可变注意力的计算是类似的。唯一的不同是在采样时，多尺度可变注意力需采样 LK 个点，单尺度可变注意力仅采样 K 个点。此外，在公式 3.2 中，多处出现归一化处理，其作用是提升收敛速度、防止模型梯度爆炸。

在公式 3.2 中，当 $L = 1$ ， $K = 1$ ， $W'_m \in \mathbb{R}^{C_v \times C}$ 为固定的单位矩阵时，可变注意力模型将退化为可变卷积。本文参考了可变卷积的思路，但可变卷积只能处理单尺度特

征图，然而，由于引入了 ResNet 中多个层的特征图，多尺度可变注意力模型可以处理来自多尺度特征图输入的多个采样点。当采样点遍历所有位置时，多尺度可变注意力模型就等同于原版的注意力模型。

为了更清楚地展示多尺度可变注意力的计算方法，将之使用类 PyTorch 的伪代码表示为算法 1 所示。

算法 1 多尺度可变注意力（类 PyTorch 代码）

```
# N: Batch Size

# M: 注意力头的数量, L: 特征图层数, K: 每个参考点的注意力采样点数

# D: 全连接隐藏层维度

# x: 输入特征图, 维度为  $N * L * D$ ,  $L_x$ :  $x$  的维度

# q: 待查询元素,  $L_q$ : 查询元素的维度, ref: 参考点坐标

# spatial_shapes: 每张特征图的大小（经过卷积后的大小）

def init():

    sample = nn.Linear(D, 2 * M * L * K)

    attn = nn.Linear(D, M * L * K)

    proj_x = nn.Linear(D, D)

    proj_output = nn.Linear(D, D)

def ms_deformable_attention(x, q, ref):

    x = proj_x(x)

    x = x.view(N, L_x, M, D // M)

    offsets = sample(q).view(N, L_q, M, L, K, 2)

    weights = attn(q).view(N, L_q, M, L * K)

    weights = softmax(weights, -1).view(N, L_q, M, L, K)

    sample_loc = ref + normaliaze(offsets)

    output = deform_attn_func(x, sample_loc, weights)

    output = proj_output(output)
```

算法 1 中， $q \in \mathbb{R}^{N \times T \times D}$ 带位置信息的特征图， T 为所有特征图序列化后的维度总和。 ref 是参考点坐标，每张特征图的 ref 都是通过是将一个与其大小相等的二维等差数列展平后分别除以特征图的大小获得。算法的工作原理为：首先将 x 经过一个线性投影，再将其维度变换为 $(N, L_x, M, D // M)$ ；之后让 q 通过两个线性层，分别将

之转换为偏移量 $offsets$ 和注意力权重 $weights$, 再将参考点 ref 与偏移量 $offsets$ 叠加得到采样点 $sample_loc$, 随后将经过一个线性层的原始数据 x 、采样点 $samp_loc$ 、使用 $softmax$ 得到的注意力权重 $weights$ 送入 $deform_attn_func()$ 函数计算相似度; 最后通过一个线性层获取最终结果。需要说明的是, 在计算相似度时, 不使用 q 的原因是 q 是带位置信息的特征图数据, 而 x 是原始特征图数据。

3.5 可变编码器

在 DETR 的编码器中, 使用了多头注意力模块, 在本文中, 将之换成为本文所提出的多尺度注意力模块以处理多尺度特征图。如图 3.4 所示, 在特征提取模块中, 选用 ResNet 50 中最后三层特征图 C3、C4 和 C5, 分别经过卷积核 1×1 、步长为 1 的卷积操作, 得到多尺度特征图 $\{x^l\}_{l=1}^{L-1} (L=4)$, C5 经过卷积核 3×3 、步长为 2 的卷积操作, 得到 x^4 。默认情况下, 每个特征图的通道数均为 $C=256$ 。

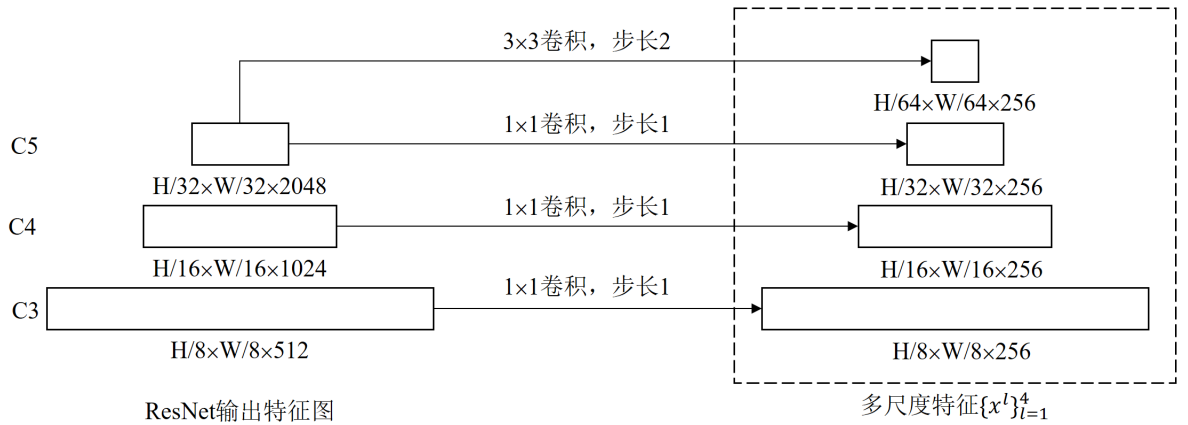


图 3.4 通过 backbone 后提取的特征图形状

在编码器中应用多尺度可变注意力模型时, 输出是与输入分辨率相同的多尺度特征图, 关键字 key 和查询 $query$ 都是多尺度特征图中的像素。对于每个查询 $query$, 它都给自己标记参考点, 也即参考点可学的。同时, 为了确定每个查询 $query$ 像素处于哪个特征图层, 需要额外增加位置信息。多尺度位置嵌入初始化是随机的, 并且是可训练的, 用 $\{e_l\}_{l=1}^L$ 来表示。

3.6 可变解码器

在解码器中存在交叉注意力与自注意力两种机制, 它们的查询元素都是目标 $query$ 。在交叉注意力模块中, 目标 $query$ 从特征图提取特征, 其中 key 元素是编码器的输出特征图。在自注意力模块中, 由于 key 元素是目标查询, 因此目标 $query$ 相互查询、彼此作用。

在 MSD DETR 中，由于解码器的输入是多尺度特征，因此使用多尺度可变注意力机制替换 DETR 中的原交叉注意力机制，自注意力机制保持不变。对于每一个目标 query，其参考点 $\hat{p}^q$ 为归一化坐标，是从目标 query 嵌入中预测得到，该预测是通过一个可学习的线性投影来得到的，激活函数是 sigmoid 函数。

如前所述，多尺度可变注意力机制从参考点附近提取图像特征，而非整张图像。因此，在做预测时会根据相对偏移量来进行预测，这样做的优点是可以降低模型的学习难度。初始化时，选择参考点作为预测框的中心点，然后，根据偏移量对预测框框的位置和大小进行调节，这样做能够加强交叉注意力与预测框的相关性，有利于提升模型的收敛速度。

3.7 预测层

预测层和 DETR 的预测层一致，由一个 3 层前向网络（Feed-Forward Networks, FFN）和一个线性投影层组成，其中 FFN 使用 ReLU 作为激活函数，隐藏层的维度为 $d=256$ 。FFN 可以预测物体边界框的中心坐标、高度和宽度，而线性投影层使用 softmax 函数用于预测查询框的类标签。在每张图中，预测固定数量的边界框，其数量记为 N ，通常 N 远大于图像中包含的实际物体数量，本文中默认 $N=300$ 。分类层的类别数为 $256+1$ （1 为背景类）。

4 实验及讨论

本章介绍目前用于目标检测常用的数据集，详细阐述实验软硬件环境，通过与其他主流方法进行实验对比，对实验结果进行分析和讨论。

4.1 数据集

目前，世界范围内较公认的数据集有 VOC^[45]、ImageNet^[46]和 COCO^[47]。

4.1.1 PASCAL VOC

PASCAL VOC（以下简称为 VOC）^[45]全称为 Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes，是目标检测常用的一个数据集，在 2005 年至 2012 年期间，每年都会举办相关的比赛。

目前，VOC 数据集主要包括 VOC 2007 和 VOC 2012 两个。其中，VOC 2012 包含 1 万余张带标签的训练和验证图像，共有 4 大类，20 个小类，具体类别如下：

人物类：人。

动物类：鸟，猫，奶牛，够，马，绵羊。

交通工具：飞机，自行车，船，公交车，汽车，摩托车，火车。

室内物体：瓶子，椅子，餐桌，盆花，沙发，电视机。

VOC 数据集的主要问题是类别过少，常被用作是目标检测方向的一个基准数据集。其所有图片都有目标检测所需要的标签，也有部分数据有语义分割的标签。

VOC 2007 中的图片被划分为训练集、验证集、测试集三部分，共有 9963 张具有标签的图片，图像中标出的总物体个数为 24640 个。

VOC 2012 中的图片分类与 VOC2007 中相同，增加了 2008 年至 2011 年所搜集到的图片，其训练、验证集、测试集总计包含 11540 张图片，共标出 27450 个物体。

4.1.2 ImageNet

ImageNet 数据集^[47]是目前世界上最大的图像识别数据库，是由美国斯坦福大学计算机科学家李飞飞建立的。最初，ImageNet 数据集用于图像识别和分类，由于其包含的图像数量巨大，主要想用于解决机器学习中的过拟合问题。

在 ImageNet 中，最常用的是 ILSVRC2012 数据集，该数据集包含 1000 个类别，称为 ImageNet 1K，共包含 120 万张训练图片，5 万张验证图片，10 万张测试图片。其中，带有目标检测标签的数据集中有 50 万张图片，共包含 200 类物体。因为该数据集过于庞大、类别繁多，导致计算量大，所以对计算环境要求很高。

后来,在此基础上进行扩充,将图像类别扩充到 2.2 万,因此也称为 ImageNet 22K,共包含近 1500 万张图片,每张图片都经过严格的人工筛选并作分类标记。

ILSVRC 全称为 ImageNet Large Scale Visual Recognition Challenge,从 2010 年至 2017 年每年举办一次,其竞赛目标包含图像分类、目标检测、视频目标检测、目标定位、场景分类等。大家所熟知的 AlexNet、VGG、GoogleNet、ResNet 等深度学习网络模型都是该项竞赛的常客。

4.1.3 Microsoft COCO

Microsoft COCO（以下简称为 COCO）^[47]全称为 Common Objects in COntext,该数据集由微软公司所建立,是计算机视觉处理最受关注、最具有权威性数据集之一。

COCO 以场景理解为目标,构建了包含有目标检测、实例分割任务的大型图像数据集。COCO 数据集共包含 32.8 万张图片,标记有 250 万个标签。

对于目标检测任务而言,COCO 数据集分为 12 个大类,80 个小类,训练和验证集共 12 万张图片,测试图片 4 万多张。与 VOC 相比,COCO 拥有更多的种类和更多的图片,并且图片中包含更多的小目标,与本文需要解决的问题符合度更高,因此在实验中选用 COCO 数据集作为实验数据集。

4.2 实验配置

4.2.1 实验软硬件环境

实验使用的计算机处理器为 Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz,运行内存 64GB, GPU 为 NVIDIA GeForce RTX 4090ti,显存 24GB,操作系统为 Windows 10 专业版。实验用 IDE 为 pycharm 2022 community edition,Python 版本为 3.7.7,Pytorch 版本为 1.12.0,Torchvision 版本为 0.13.0,Torchaudio 版本为 0.12.0,CUDA 版本为 11.6,cuDNN 版本为 8.9.0。

4.2.2 实验参数设置

实验中,网络的 backbone 部分选择在 ImageNet 22K 上进行预训练的 ResNet50,选取 C3、C4、C5 以及经 C5 处理后得到 C6 作为多尺度输入特征。可变注意力的注意力头数 (M) 设置为 8,每个参考点的注意力采样点数 (K) 设置为 4。在训练中,设置可预测的边界框最大数目 $N=300$,编码器与解码器的层数均设置为 6,预测层中隐藏层维度设置为 256,位置编码采用固定位置编码正弦 sine 函数。

训练中,使用 Adam 作为优化器,其初始化参数为,训练的轮数 epoch 设置为 50,学习率初始化为 2×10^{-4} ,当学习 40 轮后设置学习率为 2×10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.99$ 。

4.3 实验指标

AP（Average Precision）是衡量目标检测算法的最常用的指标，该指标的计算依赖于 Precision-Recall 曲线，Precision 表示检测的目标物体中是真正物体的概率，Recall 表示图像中所有目标被正确检测到的概率。通过设置不同的阈值，Precision-Recall 曲线呈现出一条曲线，AP 指标即表示 Precision-Recall 曲线下的面积。AP 指标的取值范围在 0~1 之间，值越大，表示检测算法性能越优秀。当 AP 等于 1 时，说明所有目标均被正确检测出来，其检测出的目标都是真正的物体，模型性能最优。

AP₅₀ 是指以 0.5 为 IoU（Intersection over Union，其含义为检测结果的矩形框与样本标注的矩形框的交集与并集的比值）临界值所估计出平均准确度。AP₇₅ 是指以 0.75 为 IoU（Intersection over Union，其含义为检测结果的矩形框与样本标注的矩形框的交集与并集的比值）临界值所估计出平均准确度。AP_S、AP_M 和 AP_L 分别是指图像中小型目标、中型目标和大型目标的平均准确度。

4.4 实验结果

4.4.1 与其他目标检测算法对比实验

为了验证 MSD DETR 模型的目标检测效果，我们将之与主流的目标检测算法进行对比，对比算法包括两阶段目标检测算法 Faster R-CNN、单阶段目标检测算法 YOLO 和基于 Transformer 结构的 DETR。如表 4.1 所示，Faster R-CNN 表示使用原始的 Faster R-CNN^[15]版本，backbone 选用 ResNet 50；Faster R-CNN+FPN 表示在 Faster R-CNN^[15]引入 FPN^[16]；YOLO 5s 和 YOLO 5m 分别表示 YOLO V5^[48]小规模 and 中等规模模型；DETR^[22]是以 Transformer 解码器和编码器构成的目标检测算法；本文算法及 MSD DETR 模型。

表 4.1 与主流目标检测算法对比实验结果

模型	轮数	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L	参数量(M)	FLOPs
Faster R-CNN	/	39.0	60.5	42.3	21.4	43.5	52.5	166	320
Faster R-CNN + FPN	109	42.0	62.1	45.5	26.6	45.4	53.4	42	180
YOLO 5s	300	37.4	57.2	40.2	21.1	42.3	48.9	7	16
YOLO 5m	300	53.7	73.4	58.5	34.2	59.4	74.9	21	49
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41	28

实验结果如表 4.1 所示。MSD DETR 与 DETR 相比 AP 值提升了 3.6%，尤其是在小目标检测的性能提升了 6.5%，这与 MSD DETR 中引入了多尺度特征有一定相关性，在编码器输入特征图中加入了 ResNet 提取的多个层的多个尺度特征图，使得该模型能够更好地检测出图像中的小目标。MSD DETR 的不足之处是在大目标的检测上效果不如 DETR，降低了 1.3%，其原因可能是其采用了参考点的机制，只对参考点周围的 K 个偏移量进行采样，而非对全图所有的内容进行采样，因此容易忽略掉一些尺寸较大的目标。从收敛速度看，MSD DETR 由于采用了参考点机制的可变注意力，而非使用自注意力，极大程度降低了算法复杂度，模型收敛速度较 DETR 提升了 10 倍。

与 Faster R-CNN+FPN 相比，由于二者都使用了多尺度特征，MSD DETR 在多个指标上性能均能做到领先。AP 值提升了 3.6%，小目标、中目标和大目标 AP 值分别提升了 0.5、1.5 和 5.4%。同时，MSD DETR 的网络结构比 Faster R-CNN + FPN 更加简单，参数为后者的 1/4。Faster R-CNN 的网络结构比较复杂，需要进行多个阶段训练，过程非常繁琐，而 MSD DETR 可以进行真正的端到端训练，网络参数收敛的轮数更少。

与 YOLO V5 系列中，小模型 YOLO5s 性能表现一般，随着模型规模的增大，使用 YOLO 5m 时，性能有了显著的提升，在各项性能指标比本文算法都有一定的优势，甚至在小目标 AP 上相比 MSD DETR 也有较大的优势。然而，YOLO V5 的不足之处在于其训练收敛速度较慢，需要 300 轮才能收敛，这一点不如 MSD DETR。

4.4.2 多尺度可变注意力消融实验

在 MSD DETR 中，可变注意力模块起到了极为重要的作用，表 4.2 列出了使用不同的注意力模块的设计方法进行消融实验的结果。对比表 4.2 的第 1 行和第 2 行的结果，多尺度输入对提升检测准确度效果明显，AP 值能够提升 1.7%，小目标 AP 值提升值为 2.9%，效果尤为显著。对比表 4.2 的第 2 行和第 3 行的结果，增加采样点的数量 K 也能够提升检测效果，AP 值提升了 0.9%。对比表 4.2 的第 3 行和第 4 行的结果，使用多尺度可变注意力机制后，各个层级的特征图能够进行信息融合，带来了额外的 3.3%AP 值的增长。

表 4.2 多尺度可变注意力消融实验结果

多尺度输入	多尺度注意力	<i>K</i>	AP	AP ₅₀	AP ₇₅	AP _s	AP _M	AP _L
否	否	1	39.7	60.1	42.4	21.2	44.3	56.0
是	否	1	41.4	60.9	44.9	24.1	44.6	56.1
是	否	4	42.3	61.4	46.0	24.8	45.1	56.3
是	是	4	45.6	63.9	47.5	27.1	46.9	58.8

4.4.3 编码器层数消融实验

在 MSD DETR 编码器中，使用了多尺度可变注意力模块，本实验改变多尺度可变注意力模块的层数，其结果如表 4.3 所示。观察表 4.3 可得出结论，随着编码器的增加，目标检测的效果整体呈上升的趋势，尤其当编码器层数较少时，上升趋势尤为明显。例如，当编码器层数从 2 上升到 4 时，AP 值上升了 6.4%。当编码器层数较多时，增加层数的效果就不再显著。例如，当编码器层数从 6 上升到 12 时，AP 值上升仅为 0.4%，且对于小目标的检测效果，反而出现了 0.1%的下降。同时，随着编码器层数的增加，网络参数量也有显著的增长，当编码器层数从 6 上升到 12 时，参数增长了 8.8M，这对于网络的收敛提出了更大的挑战。

表 4.3 编码器层数消融实验结果

编码器层数	参数量(M)	AP	AP ₅₀	AP ₇₅	AP _s	AP _M	AP _L
2	37.9	38.5	58.3	43.7	21.9	41.6	56.3
4	38.5	44.9	62.5	46.0	25.2	43.8	57.9
6	40	45.6	63.9	47.5	27.1	46.9	58.8
12	48.8	46.0	64.1	48.5	27.0	47.1	59.2

4.4.4 解码器层数消融实验

在 DETR 解码器中，使用了自注意力模块和交叉注意力模块，而在 MSD DETR 中，使用多尺度可变注意力模块替换了交叉注意力模块，保留自注意力模块。本实验改变注意力模块的层数（每 1 层注意力由 1 个多尺度可变注意力模块和 1 个自注意力模块构成），其结果如表 4.4 所示。

观察表 4.4 可得出结论，随着解码器层数的增加，目标检测的效果整体呈上升的趋势，但这种上升趋势并不明显，与增加编码器层数带来的性能提升相比，增加解码器层数只能小幅度提升目标检测的性能，从 2 层增加到 12 层仅提升了 1.4%，甚至在

一些指标上，还出现了小幅下降，当编码器层数从 6 上升到 12 时，对于大目标的检测效果，反而出现了 0.2% 的 AP 值下降。

表 4.4 解码器层数消融实验结果

编码器层数	参数量(M)	AP	AP ₅₀	AP ₇₅	AP _s	AP _M	AP _L
2	37.9	44.3	61.9	44.8	23.8	43.7	55.9
4	38.5	45.9	63.5	46.9	26.7	45.8	56.2
6	40	45.6	63.9	47.5	27.1	46.9	58.8
12	48.8	45.7	64.0	47.6	27.2	47.1	58.6

4.5 可视化实验

为了更加直观地观察本文算法与其他算法在目标检测中的效果差异，本实验将实验结果进行展示。图 4.1-图 4.4 为使用本文算法和使用 Faster R-CNN + FPN 模型检测效果的对比，检测出的物体均用边界框标注。

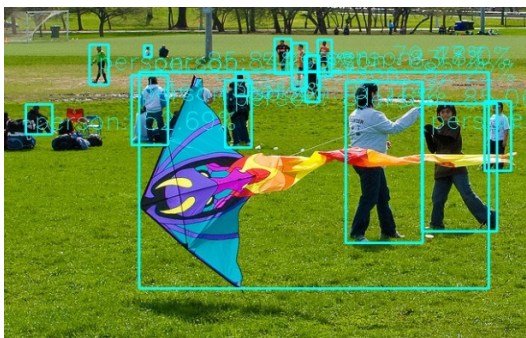


图 4.1 MSD DETR 检测效果

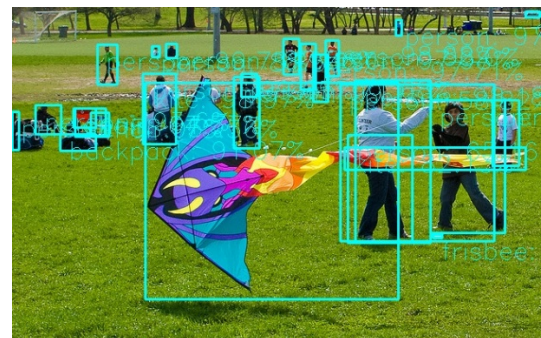


图 4.2 Faster R-CNN+FPN 检测效果

对比图 4.1 与图 4.2 分析得出，MSD DETR 检测出的边界框数量少于 Faster R-CNN+FPN 检测的边界框，且边界框质量更高、贴近于真实目标。如图 4.1 中，MSD DETR 能够检测出包括尾部在内的完整风筝，而 Faster R-CNN 错误地将风筝检测为两个物体。主要原因是 Faster R-CNN 在检测时先使用 RPN 选取候选区域，此时风筝主题和尾部被划分到不同的候选区域；而 MSD DETR 没有提取后全区域的步骤，而且其中 Transformer 模型的全局信息获取能力强于卷积神经网络。此外，一般而言，选取候选区域能够有效地防止漏框的情况出现，但边界框也非越多越好。如图 4.3 与 4.4 所示，同样是一只猫，Faster R-CNN 误将图像检测出两个边界框，分别为一个人和一只猫，这是由于边界框过于依赖分类准确度而造成的结果。



图 4.3 MSD DETR 检测效果

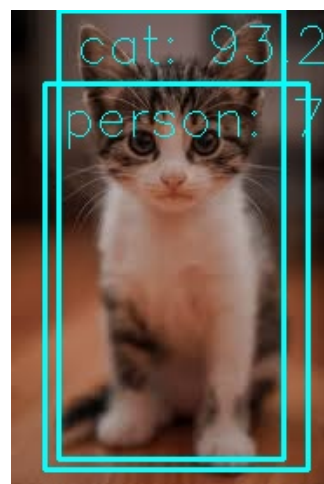


图 4.4 Faster R-CNN+FPN 检测效果

5 结论与展望

对注意力机制和目标检测问题进行研究，在 DETR 的基础上，提出了一种多尺度可变 DETR 模型。该模型能够支持端到端、单阶段的目标检测方法，能够很好地对图像中各种大小的目标进行高效、准确的检测。

5.1 结论与分析

本文主要探讨了 Transformer 在目标检测领域的应用，虽然它才刚刚进入视觉领域，但基于它设计的 ViT 模型在图像分类上的成绩是非常瞩目的，日后它也会在计算机视觉领域取得更显著的结果。

关于本文的工作，我们首先详细的介绍了目标检测的发展历程，并在之后对目标检测的意义进行了分析。

其次我们简述了注意力机制存在的问题，并讲述了现阶段解决这些问题的基本思路。然后的分析了注意力机制，以及多头注意力的计算过程。与传统的卷积相比注意力机制更加注重全局信息，能够更好的捕获图像的全局信息。

在之后，我们讲解了 Deformable DETR 的网络结构，并且对其核心机制——可变注意力机制的计算过程进行了详细的解释，并且对可变注意力做了详细的解释，与原版的 Transformer 相比，我们在编码器中将自注意力替换成了可变注意力，将解码器的交叉注意力替换成了可变注意力。

最后，我们通过与原版的 DETR、YOLOv5、Faster R-CNN+FPN 进行对比，我们发现相比 DETR 已经有了很大的进步，但是与 YOLOv5 相比，特别是 YOLOv5 的中大型模型相比仍旧是有很大的差距，不过我们并不灰心，因为 Transformer 才刚刚进入目标检测领域，我们还有很长的路要走，我们相信 Transformer 在目标检测中必会大放光芒。与成名已久的 Faster R-CNN + FPN 相比我们的 Deformable DETR 与其性能相差不大，模型结构更为简单，训练过程更为简便。

综上所述，本文在 DETR 上的修改是较成功的，即使比不过成名已久的 YOLO 网络，但这次更改也为后续的改进提供了一些有用的思想。这也说明了 Transformer 在目标检测领域是很有发展前途的，但对比已经发展起来的 YOLO 来说仍旧需要一些时间来改进。Deformable DETR 刚刚出世就已经可以比拟甚至略微超过了同样使用多尺度特征图的 Faster R-CNN + FPN，这也说明了 Transformer 在目标检测领域的发展潜力。Transformer 在目标检测领域仅仅是崭露头角便以锋芒毕露，我相信其未来在目

标检测领域，甚至是整个计算机视觉领域都会有更进一步的发展，并且带领计算机视觉处理进入一个新的层次。

5.2 展望未来

随着深度学习的发展，计算机视觉领域也在进一步发展。越来越多的想法、技巧被应用到目标检测的检测器中，就如 DETR 一般，在 ViT 之后就将 Transformer 引入了目标检测领域。

但是 DETR 在将 Transformer 引入目标检测之后并没有在用一些技巧来提高准确度或者检测速度。我想这是未来吸引更多的学者来研究 Transformer 在图像领域的应用吧。

所以为了加强 Transformer 在目标检测领域的影响，我们可以在如下的方向做一些努力。

首先可以对注意力机制做出更改，就如注意力机制的变种，通道注意力、空间注意力、时间注意力、分支注意力等等，都是在注意力机制方面做出了改变。注意力的本质是将有限的注意力集中在最有用的信息上面，节省资源，迅速获取有效信息，或许我们可以提高其获取信息的质量。我们此次也是在参考了 FPN 结构与可变卷积之后才想到为什么不能让注意力机制也支持多尺度特征图呢？最终我们提出了可变注意力模型，它能够很好的处理多尺度特征图。另外我想我们可以进一步的探索基于注意力机制的预训练模型，或者是设计出一种高效且通用的注意力模型来匹配各类具体的任务。

其次，我们可以优化模型的结构，对模型的训练参数进行优化求解，得出最优模型，并进行训练。就如本文提出的 Deformable DETR 我们仍然可以对其模型进一步优化，比如对其进行双阶段训练。

最后，我们可以联合各个领域学者，加强学术交流，比如 Transformer 最早是应用在语言领域的，如果能够将两个领域甚至多个领域的研究进行整合提取对各自有利的信息，我相信 Transformer 的发展会更加迅速。

Transformer 在刚刚出世的时候就在自然语言处理领域展现了其强大的全局信息处理能力，而在进入计算机视觉领域后也很快带起来了计算机视觉领域对 Transformer 的研究。Transformer 的发展潜力是毋庸置疑的，而它的核心是注意力机制，我读到过一篇文章很形象的将注意力机制在深度学习中的作用体现出来了。文章将人类的学习分为四个阶段，分别是：死记硬背阶段；提纲挈领阶段；融会贯通阶段；登峰造极阶

段。

在 RNN 时代，模型就仍旧处于死记硬背的阶段，而注意力机制使得模型学会了提纲挈领，从而进化到 Transformer 的融会贯通阶段，能够从复杂的图像或者对话中理解上下文信息，并且举一反三的进行学习。

在之后的研究中，我们可以从 Transformer 本身的结构进行分析研究，通过优化 Transformer，或提升其性能，或更改其模型使其更加高效，或使其能够适用于更多的任务，又或者是从注意力机制上寻求突破。

结束语

本文依托江苏省高校自然科学研究面向项目“面向砂岩图像的图像全景分割技术研究”（编号：19KJB520038）和江苏省高等学校大学生创新创业训练计划立项项目“基于 Transformer 的图像识别方法的研究”（编号：202213842037Y），对注意力机制和目标检测问题进行研究，在 DETR 的基础上，提出了一种多尺度可变 DETR 模型，能够很好地对图像中各种大小的目标进行高效、准确的检测。

本文主要围绕 DETR 模型展开，在 DETR 的基础上进行优化。DETR 是 Facebook 团队在三年前提出的基于 Transformer 的端到端目标检测模型，没有非极大值抑制 NMS 后处理步骤，没有 anchor 等先验知识的约束，大大简化了目标检测的流程。但是 DETR 的缺点也很明显，首先，由于 Transformer 模型结构本身的缺陷，DETR 的收敛速度很慢。其次就是能够处理的特征分辨率有限，与现在的 FPN 的多尺度特征处理相比还有些不足。所以本文基于这两个点提出了可变注意力机制，替换了 DETR 中编码器的自注意力机制，模型的具体实现在第三章都有详细说明。我们相信可变注意力机制将会在目标检测领域展现出其强大的推理能力。

致 谢

首先，感谢姜枫老师在毕业设计期间对于文献查找与阅读、科研方法、论文写作技巧和规范方面对我耐心的教导，让我能够顺利完成本篇论文。然而一篇毕业论文的完成并不能说明我对人工智能算法的探索结束了，在未来的日子里，我会倍加努力的学习人工智能前沿技术，争取在人工智能领域开拓进取，学以致用，有朝一日能够有所成就，不负姜老师的教诲。

其次，感谢所有在学业上对我有所帮助的老师以及同学们，在程序设计以及算法方面对我耐心的解答，给出了宝贵的意见，帮助我消除疑惑，让我能够继续奋力向前。

感谢父母家人对我的养育之恩，二十年来对我的支持。

感谢学界泰斗们无私的奉献，无偿的分享研究成果，让我能够接触到人工智能的前沿算法。

最后感谢一下自己，即便人生艰难，也坚持到现在。以后的日子里，无论是继续深造，还是参加工作，都能够坚持到底；愿我付出甘之如饴，所得皆为尾所愿；愿我活成自己想成为的模样，去靠近美好，进而成为美好本身。

参 考 文 献

- [1] 罗金,王丽珍,王晓璇等.k-近邻关系下的空间高效用核模式挖掘[J].计算机学报,2022,45(02):354-368.
- [2] 李宇溪,周福才,徐紫枫.支持 K-近邻搜索的移动社交网络隐私保护方案[J].计算机学报,2021,44(07):1481-1500.
- [3] 谭雪敏,吴远峰,袁正午等.拉格朗日多项式逻辑回归分类算法并行计算优化[J].遥感信息,2016,31(01):96-101.
- [4] 郭华平,董亚东,邬长安等.面向类不平衡的逻辑回归方法[J].模式识别与人工智能,2015,28(08):686-693.
- [5] Mohammed H. Affirmative Ant Colony Optimization Based Support Vector Machine for Sentiment Classification[J].Electronics,2022,11(7).
- [6] 周开伟,钱雪忠,周世兵.一种改进的多分类孪生支持向量机[J].计算机应用与软件,2022,39(04):269-274+299.
- [7] 高虹雷,门昌骞,王文剑.一种特征值区间划分的模型决策树加速算法[J].小型微型计算机系统,2021,42(06):1136-1143.
- [8] 唐亮,李飞.基于决策树的车联网安全态势预测模型研究[J].计算机科学,2021,48(S1):514-517.
- [9] 关晓蕾,王文剑,庞继芳等.基于空间变换的随机森林算法[J].计算机研究与发展,2021,58(11):2485-2499.
- [10] 刘峻.基于单图像超分辨率的约束随机森林算法[J].计算机工程与设计,2017,38(04):970-975.
- [11] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014:580-587.
- [12] Uijlings J, Sande K, Gevers T, et al. Selective search for object recognition. International Journal of Computer Vision, 2013, 104(2):154-171.
- [13] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904-16.
- [14] Girshick R. Fast R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2016.
- [15] Ren S, He, K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(6):1137-1149.

-
- [16] Lin T, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [17] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [18] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [19] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [20] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector [C]// European Conference on Computer Vision (ECCV), 2016.
- [21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]// Advances in Neural Information Processing Systems, 2017:5999-6009.
- [22] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]// European Conference on Computer Vision (ECCV), 2020:213-229.
- [23] Ahmed, I M, Maalej R, Kherallah M. MobileNet-Based Model for Histopathological Breast Cancer Image Classification[J]. Lecture Notes in Networks and Systems, 2023, 647:636-643.
- [24] Shah P, El-Sharkawy M. A-MnasNet: Augmented MnasNet for Computer Vision [C]// Midwest Symposium on Circuits and Systems, 2020:1044-1047.
- [25] Wang P, Yang A, Men R, et al. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework[C]// International Conference on Machine Learning (ICML), 2022, 162:23318-23340.
- [26] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. IEEE Conference on Computer Vision and Pattern Recognition. 2014:580-587.
- [27] Child R, Gray S, Radford A, et al. Generating long sequences with sparse transformers[J]. arXiv preprint arXiv:1904.10509, 2019.
- [28] Beltagy I, Peters M E, Cohan A. Longformer: The long-document transformer [J]. arXiv preprint arXiv:2004.05150, 2020.
- [29] Kitaev N, Kaiser L, Levskaya A. Reformer: The efficient transformer[J]. arXiv preprint arXiv:2001.04451, 2020.
- [30] Wang S, Li B Z, Khabsa M, et al. Linformer: Self-attention with linear complexity[J]. arXiv preprint arXiv:2006.04768, 2020.
- [31] Alexey D, Lucas B, Alexander K, et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale[C]// International Conference on Learning Representations, 2021.
- [32] Zhu X, Cheng D, Zhang Z, et al. An empirical study of spatial attention mech

- anisms in deep networks[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6688-6697.
- [33] Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2017:764-773.
- [34] Wu F, Fan A, Baeviski A, et al. Pay less attention with lightweight and dynamic convolutions[J]. arXiv preprint arXiv:1901.10430,2019.
- [35] Ramachandran P, Parmar N, Vaswani A, et al. Stand-alone self-attention in vision models[J]. arXiv, 2019.
- [36] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
- [37] Yao Q, Fan X, Gong M. PANet: A Point-Attention Based Multi-Scale Feature Fusion Network for Point Cloud Registration[J]. IEEE Transactions on Instrumentation and Measurement. 2023, 72: 414-427.
- [38] Kong T, Sun F, Tan C, et al. Deep feature pyramid reconfiguration for object detection[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 169-185.
- [39] Zhao Q, Sheng T, Wang Y, et al. M2det: A single-shot object detector based on multi-level feature pyramid network[C]// AAAI conference on artificial intelligence. 2019, 33(01): 9259-9266.
- [40] Ghiasi G, Lin T Y, Le Q V. Nas-fpn: Learning scalable feature pyramid architecture for object detection[C]// IEEE/CVF conference on computer vision and pattern recognition. 2019: 7036-7045.
- [41] Xu H, Yao L, Zhang W, et al. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6649-6658.
- [42] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2016:770-778.
- [43] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Proceedings of the IEEE on Computer Vision and Pattern Recognition (CVPR). 2014:580-587.
- [44] Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks[C]// Processing of IEEE International Conference on Computer Vision (ICCV). 2017:764-773.
- [45] Everingham M, Gool LV, Williams CKI, et al. The pascal visual object classes (VoC) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.
- [46] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015,115(3):211-252.

- [47] Lin TY, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context[C]// Proceedings of the European Conference on Computer Vision (ECCV). 2014: 740–755.
- [48] John D. YOLOv5: a better version of YOLO[J]. IEEE Transactions on image processing, 2021, 30(5):1234-1245.