基于 Transformer 的端到端对象检测

摘要

本文提出了一种新的目标检测方法，即 DEtection TRansformer (DETR)，它将检测视为直接的集合预测问题。DETR 采用 Transformer 编码器-解码器架构，并使用基于集合的损失函数与二分匹配来确保预测的独特性。该模型通过一组学习到的对象查询，分析对象间的关系及全局图像上下文，直接并行输出最终预测结果。DETR 概念简单，无需专用库或手工设计组件如非极大值抑制或锚点生成。在 COCO 数据集上，DETR 的性能可与优化后的 Faster R-CNN 基线相媲美，并且易于扩展至全景分割任务，展现出显著优势。

关键词：目标检测；集合预测；DETR；二分匹配

1 引言

目标检测任务旨在预测每个感兴趣物体的边界框和类别标签。传统的目标检测方法通常通过间接的方式解决这一问题，使用回归和分类任务处理大量的候选框 [3]、锚框 [9] 或窗口中心 [19]。然而，这些方法的性能受到后处理步骤的显著影响，如合并重复预测、锚框集合的设计和将目标框分配给锚框的启发式方法 [18]。因此，如何简化目标检测流程并提高模型的精度和效率，一直是计算机视觉领域的研究热点。

为了解决这些问题，提出了端到端的目标检测框架，其中一个重要的方向是通过直接集合预测来简化传统的检测流程。DETR 作为这一方向的代表性工作，采用了基于 Transformer [15] 的编码器-解码器架构，摒弃了锚框设计、非极大值抑制等传统组件，采用全新的直接集合预测方式。DETR 通过自注意力机制能够显式建模序列中各个元素之间的相互关系，从而有效去除重复预测并提高检测精度。此外，DETR 使用的二分匹配损失函数能够将预测结果与真实目标进行一一对应，使得模型的训练更加简洁高效。

DETR 的创新设计使其在大目标检测上表现出显著的优势，尤其是变换器的非局部计算能力帮助其更好地捕捉远距离上下文信息。然而，DETR 在小目标检测上的性能仍然存在一定提升空间。通过与 Faster R-CNN 等基准模型 [12] 的对比实验，DETR 在大目标上的表现优于传统方法，展示了其在复杂目标检测任务中的潜力。此外，DETR 的架构不仅适用于目标检测任务，还能够扩展到全景分割 [7] 等复杂视觉任务，为未来计算机视觉研究提供了新的思路和方向。

DETR 通过基于变换器（transformer）的端到端架构，摒弃了传统目标检测方法中复杂的后处理步骤，如锚框设计和非极大值抑制，直接通过自注意力机制进行集合预测。这种方法简化了检测流程，并在处理大目标和复杂背景时展现了显著的优势。尽管如此，DETR 在小目标检测上的性能仍然存在一定的提升空间，因此，针对 DETR 进行优化，特别是在小目

标检测方面，具有重要的研究价值。随着计算资源和硬件技术的发展，DETR 的架构在提高检测精度和效率方面具有较大的潜力。选择 DETR 作为研究对象，能够推动计算机视觉技术向更加高效、精准的方向发展。

2 相关工作

本文的工作建立在几个领域的先前工作基础之上：用于集合预测的二部匹配损失、基于 Transformer 的编码器-解码器架构、并行解码以及目标检测方法。

2.1 集合预测

基本的集合预测任务是多标签分类 [11, 14]。对于该任务，基线方法（一对多）不适用于诸如检测之类的存在元素之间潜在结构（几乎相同的框）的问题。在这些任务中，第一个难点是避免近重复。大多数当前的检测器使用后处理步骤，如非最大抑制（NMS），来解决这个问题，但直接的集合预测不依赖于后处理。它们需要能够建模所有预测元素之间相互作用的全局推理机制，以避免冗余。对于恒定大小的集合预测，密集的全连接网络 [4] 可以满足，但是代价高。一种通用的方法是使用自回归序列模型，如递归神经网络 [16]。在所有情况下，损失函数应对预测的排列保持不变。通常的解决方案是基于匈牙利算法设计一个损失函数 [8]，用以在真实标注和预测之间找到二分匹配。这强制实施排列不变性，并确保每个目标元素都有一个唯一的匹配。与大多数先前的工作不同，论文摒弃了自回归模型，采用了并行解码的变换器模型。

2.2 Transformer 与并行解码

Transformers 由 Vaswani 等人 [15] 提出，作为一种基于注意力的新模块用于机器翻译。注意力机制 [1] 聚合整个输入序列的信息，Transformers 引入自注意力层，类似非局部神经网络 [17]，通过聚合全序列信息更新每个元素。其全局计算和完美记忆使得 Transformers 比 RNN 更适合长序列任务，现已在自然语言处理、语音处理和计算机视觉中广泛替代 RNN。Transformers 用于自回归模型，但高推理成本促使并行序列生成的出现，应用于音频、机器翻译、词表示学习和语音识别。我们结合 Transformers 和并行解码，因其在计算成本和全局计算能力之间的良好平衡。

2.3 目标检测

现代目标检测方法通常依赖于相对于初始猜测（如锚点或提议框）进行预测，但研究 [18] 表明，系统的性能很大程度上取决于这些初始猜测的设定方式。为改进这一点，本文的模型直接针对输入图像进行绝对边界框预测，简化了检测过程并移除了手工设计的初始猜测。此外，通过使用集合损失和二分匹配损失，结合可学习的非极大值抑制（NMS）[5] 和关系网络 [6]，我们减少了模型中先验知识的编码，并避免了额外的手工设计特征。与之类似的方法还包括用于目标检测和实例分割的端到端集合预测，它们采用基于 CNN 激活的编码器-解码器架构，不过这些方法主要依赖自回归模型（如 RNN），而本文提出的方法则充分利用最新的并行解码的变压器架构，以提高效率和准确性。

3 本文方法

3.1 本文方法概述

DETR 的实现基于 Transformer 架构，通过编码器-解码器结构来直接预测目标检测的集合。编码器接收经过卷积神经网络（CNN）提取的图像特征，并通过自注意力机制捕捉全局上下文信息。解码器则接收一组固定数量的学习对象查询（object queries），并结合编码器的输出，通过自注意力和编码器-解码器注意力机制并行生成每个对象的类别和边界框预测。最终，通过集合损失函数进行训练，该函数使用二分图匹配来唯一地将预测与真实目标匹配，确保每个预测对象与一个真实对象相对应，从而避免重复预测。可以参照如图 1所示：

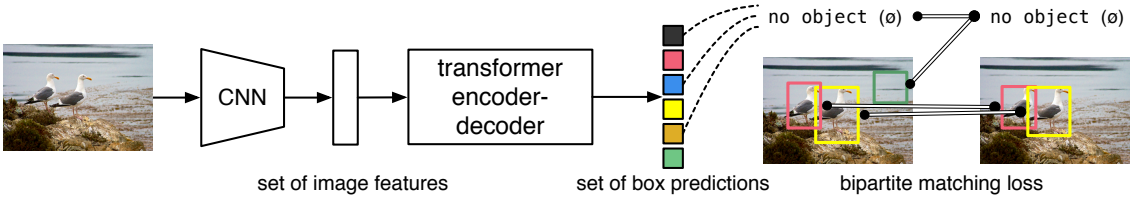

图 1. DETR 框架图

3.2 目标检测集合预测损失

DETR 在单次解码过程中生成 N 个预测，N 大于实际图像中的目标数量，损失函数生成预测对象和真实对象之间的最优二分匹配，然后优化特定于对象的边界框损失。

用 y 表示真实对象的集合，$\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ 表示 N 个预测结果的集合，用 $\emptyset$（无对象）进行填充。我们的目标是在 N 个元素排列 $\sigma \in \mathfrak{S}_N$ 中找到使两个集合之间二分匹配代价最小的排列：

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}), \tag{1}$$

其中 $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ 表示真实值 y_i 与索引为 $\sigma(i)$ 的预测值之间的匹配成本，这种最优分配通过匈牙利算法计算得出。

匹配成本同时考虑了类别预测和预测框与真实框的相似性。真实值集合中的每个元素 i 可以表示为 $y_i = (c_i, b_i)$，其中 c_i 是目标类别标签（可能是空 $\emptyset$），而 $b_i \in [0,1]^4$ 是一个定义真实框中心坐标及其相对于图像大小的高度和宽度的向量。对于索引为 $\sigma(i)$ 的预测，我们定义类别 c_i 的概率为 $\hat{p}_{\sigma(i)}(c_i)$，预测框为 $\hat{b}_{\sigma(i)}$。基于这些符号，我们定义匹配成本 $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ 如下：

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{1}_{\{c_i \neq \emptyset\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}), \tag{2}$$

匹配成本的第二部分和匈牙利损失是 $\mathcal{L}_{\text{box}}(\cdot)$，用于评估边界框。与许多通过初始猜测进行增量预测的检测器不同，我们直接预测边界框。我们使用 l_1 损失和广义 IoU 损失 [13] 的线性组合解决这个方法导致的损失的相对缩放问题。总体上，我们的边界框损失定义为：

$$\mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) = \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} \| b_i - \hat{b}_{\sigma(i)} \|_1, \tag{3}$$

其中 $\lambda_{\text{iou}}, \lambda_{\text{L1}} \in \mathbb{R}$ 是超参数。这两种损失被批次中的对象数量归一化。

下一步是计算所有匹配对的匈牙利损失，该损失函数由类别预测的负对数似然和边界框损失的线性组合构成，类似于常见目标检测器的损失定义：

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) \right], \tag{4}$$

其中，$\boldsymbol{\sigma}$ 是在第一步中计算得到的最优分配（公式 (1)）。当 $c_i = \emptyset$ 时，需要将对数概率项的权重降低 10 倍来处理类别不平衡问题。这类似于 Faster R-CNN 训练过程中通过下采样平衡正负样本的方式 [12]。对象与空集 $\emptyset$ 之间的匹配成本不依赖于预测，这种情况下的成本是一个常数。

3.3 DETR 架构

DETR 主要有三个组件：CNN 主干网络，Transformer 编码器-解码器以及前馈神经网络 (FFN)，具体架构如图 2所示。

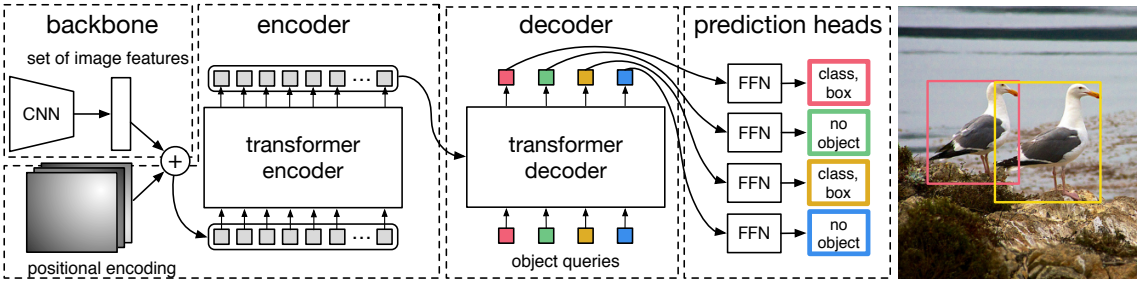

图 2. DETR 详细框架图

3.3.1 CNN 主干网络

从初始图像 $x_{\text{img}} \in \mathbb{R}^{3 \times H_0 \times W_0}$ 开始，利用传统的 CNN 主干网络生成一个低分辨率的激活图 $f \in \mathbb{R}^{C \times H \times W}$。常用的参数值是 $C = 2048$，$H, W = \frac{H_0}{32}, \frac{W_0}{32}$。

3.3.2 Transformer 编码器

先使用 1x1 卷积将高维激活图 f 的通道维度从 C 减少到较小的维度 d，生成新的特征图 $z_0 \in \mathbb{R}^{d \times H \times W}$。再将 z_0 的空间维度压缩成 $d \times HW$ 的特征图作为编码器输入。每个编码器层包含多头自注意力模块和前馈神经网络 (FFN)。由于 Transformer 结构是排列不变的，我们在每个注意力层的输入中添加固定位置编码 [2, 10]。

3.3.3 Transformer 解码器

解码器遵循标准的 Transformer 架构，使用多头自注意力和编码器-解码器注意力机制处理 N 个大小为 d 的嵌入。与原始 Transformer 不同的是，我们的模型在每个解码器层并行解码 N 个对象。这些输入嵌入是学习的位置编码，称为对象查询，并且在每个注意力层的输入中添加它们。解码器将 N 个对象查询转换为输出嵌入，然后通过 FFN 解码为框坐标和类别标签。

3.3.4 预测前馈神经网络 (FFNs)

最终预测由一个带有 ReLU 激活函数和隐藏维度为 d 的三层感知机以及一个线性投影层计算得出。FFN 预测相对于输入图像的归一化中心坐标、高度和宽度，线性层使用 softmax 函数预测类别标签。由于我们预测一组固定大小的 N 个边界框，其中 N 通常远大于图像中实际感兴趣对象的数量，因此引入了一个额外的特殊类别标签 $\emptyset$ 来表示在某个槽位内未检测到任何对象。这个类别在标准目标检测方法中扮演类似于“背景”类的角色。

3.3.5 辅助解码损失

在每个解码器层之后添加了预测 FFN 和匈牙利损失作为辅助损失，以帮助模型更好地学习每个类别的正确对象数量。所有预测 FFN 共享参数，并使用额外的共享层归一化来规范输入，确保不同解码器层的预测一致性。

4 复现细节

4.1 与已有开源代码对比

论文中提供了对应的源代码和 COCO2017 数据集。源代码使用 8 张显卡进行训练，而在复现时仅使用 1 张显卡，这导致训练时间显著延长。使用相同的参数和数据集从零开始训练时，每轮的分类误差接近 100%。研究发现，训练时所使用的显卡数量会影响学习率的设定。为了使本次实验（使用 1 张显卡）的训练效果接近源代码中多张显卡训练的日志效果，我们相应地将学习率调整为原学习率的 $\frac{1}{8}$。之后，我们使用 VOC 数据集进行测试，并将其转换为 COCO 数据集格式（转换代码自行编写）。同时，我们利用 DETR 提供的预训练模型进行实验，结果表明，随着迭代次数的增加，目标检测的分类效果逐渐提升，并达到与论文相当的效果。

在参考了本论文的开源代码（链接如下：https://github.com/facebookresearch/detr）作为复现基础后，我们进一步使用了更简洁的代码对 DETR 进行微调。参考的源码地址为：https://github.com/NielsRogge/Transformers-Tutorials/blob/master/DETR/Fine_tuning_DetrForObjectDetection_on_custom_dataset_(balloon).ipynb。为了提升基于 Balloon 数据集训练的源码的功能与性能，我们对其进行了结构上的完善与优化。首先，引入了中断控制机制，确保训练过程中能够随时保存模型状态，并支持将模型便捷地上传至 HuggingFace 仓库，极大增强了模型管理与部署的灵活性。在性能调优方面，采用了 StepLR 学习率调度器，实现学习率的动态调整，以促进模型更高效地收敛。同时，启用了混合精度训练，通过设置 Trainer 的 precision=16，有效加速了训练进程并降低了内存消耗，实验效果与原版相近。尽管在尝试引入提示词微调功能，但效果未达预期，显示出该功能尚需进一步的优化与改进，观察到其效果并不如预期般显著，这可能与提示词的设计、数据预处理方式或学习率调整有关，已提交相关源码。

4.2 实验环境搭建

本次实验的算法复现是在配备 Intel(R) Core(TM) i9-14900K 3.20 GHz 处理器的计算平台上完成的。我们基于原始 DETR 源代码进行了复现，设置初始学习率为 0.125×10^{-4}，并

采用批次大小（batch size）为 8 进行了总计 50 个 epoch 的训练。随后，使用预训练的 DETR 模型参数进一步优化模型性能。

此外，为了适应新的数据集，我们对图像进行了预处理工作，包括类别映射、文件格式从 XML 到 JSON 的转换，以及确保每个图像与其对应的标注信息准确无误地匹配。这些步骤保证了数据集的有效性和一致性，从而为模型训练提供了坚实的基础。

4.3 使用说明

在进行代码复现前，使用 voc 数据集对 DETR 论文源码进行复现时，需要运行 `voc-to-coco.py` 文件，该文件会将 voc 数据集格式转成 coco 数据集格式。或者执行提交的源码前，需要执行 VIA2COCO 文件夹下的 to.py 文件，将 balloon 数据集转成训练时需要的数据集格式。具体执行命令和文件说明请参照提交的 README.md 文件。

5 实验结果分析

使用 balloon 数据集复现 DETR 的训练损失变化曲线如图 3所示：

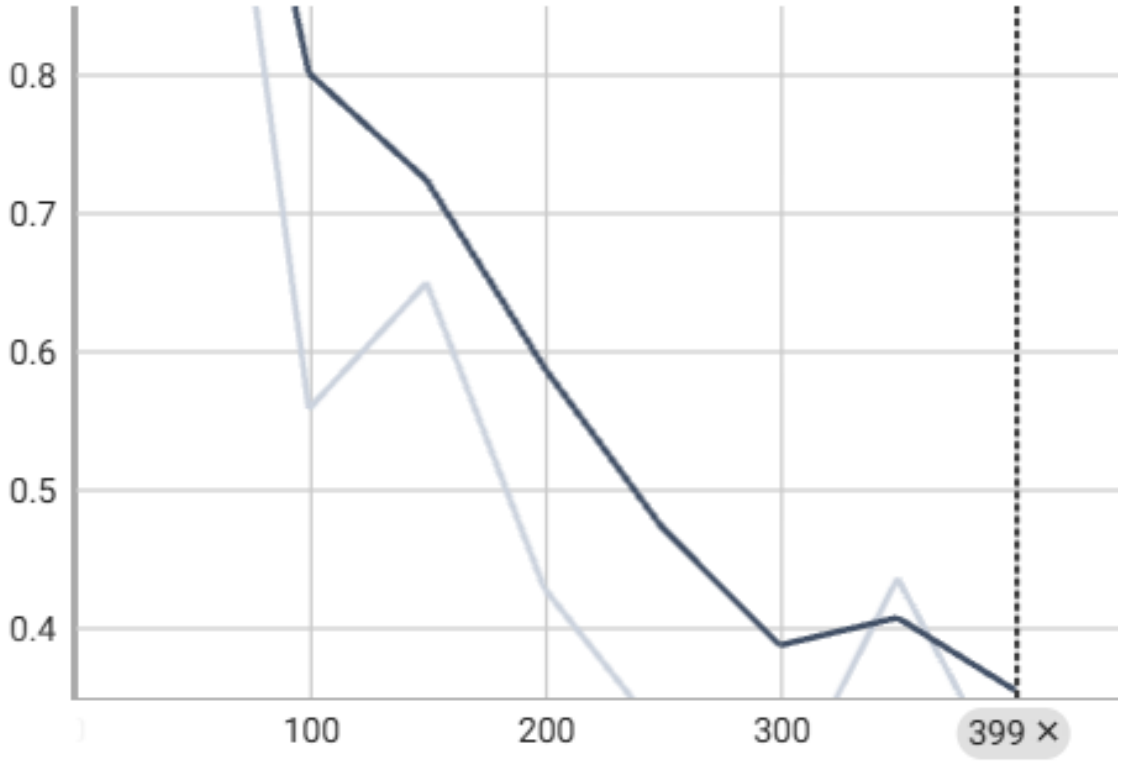

图 3. 训练损失变化曲线图

使用 balloon 数据集复现 DETR 的验证损失变化曲线如图 4所示：

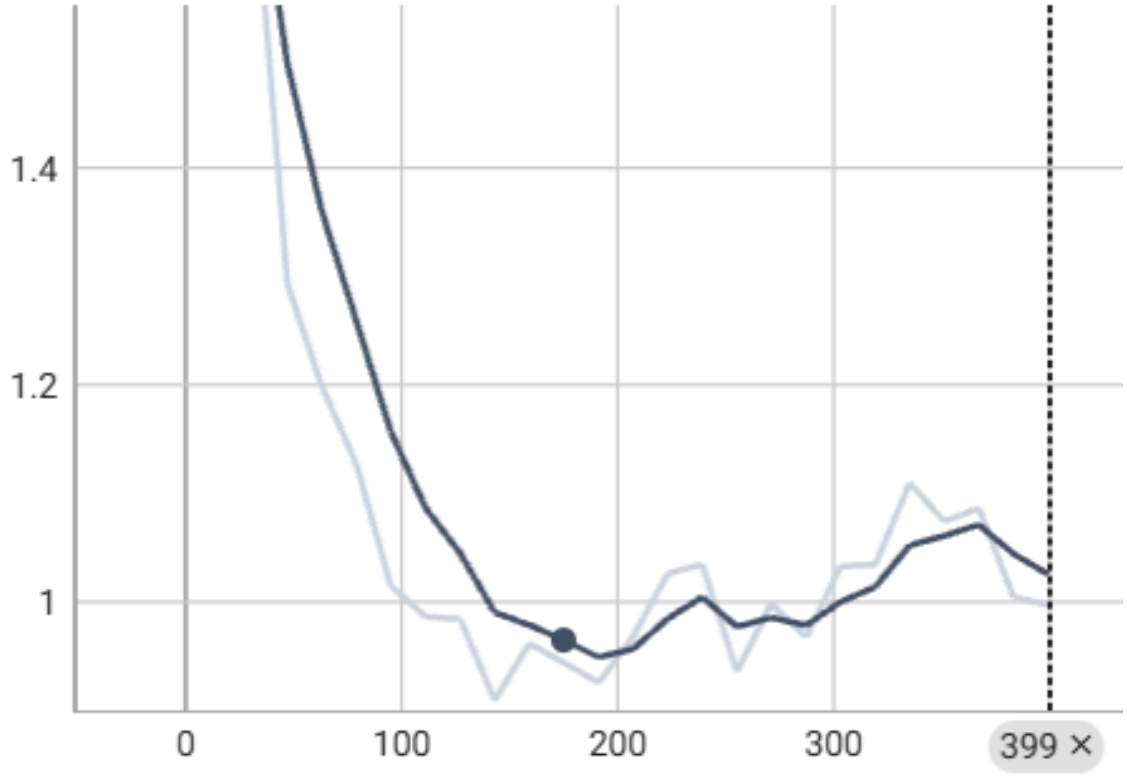

图 4. 验证损失变化曲线图

实验结果如表1所示：

表 1. 实验结果

模型	数据集	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR(原文)	COCO 数据集	42.0	62.4	44.2	20.5	45.8	61.1
DETR(复现)	COCO 数据集	41.5	61.9	44.0	20.1	45.5	60.5
DETR(复现)	VOC 数据集	55.0	69.7	60.2	23.3	36.5	65.5
DETR(复现)	balloon 数据集	51.2	76.6	53.6	44.2	74.7	–

对气球的目标检测实验结果如图 5所示：

图 5. 目标检测图

从实验结果上看，DETR 模型在不同数据集上展示了其广泛的适用性和良好的性能。具体而言，在 COCO 数据集上的复现结果显示，我们的实现与原始 DETR 模型的性能非常接近，这表明了我们复现工作的成功。特别是在平均精度（AP）方面，复现模型仅略低于原文模型，证明了在相同条件下，两个模型具有相似的表现力。

对于 VOC 数据集，DETR 复现模型表现出更高的平均精度（AP），尤其是 AP_{50} 和 AP_L 的提升明显，说明该模型在处理较为简单或大规模物体时具有较强的识别能力。这一结果进一步验证了 DETR 模型的鲁棒性和通用性。

然而，在 balloon 数据集上的表现则显示出不同的特点。尽管在 AP_{50} 上达到了较高的分数，但在小目标检测（AP_S）上的表现却相对较低。这提示我们，DETR 在处理特定的小目标检测任务时可能需要额外的优化措施，例如改进的特征提取机制或更有效的数据增强策略。

此外，通过图 5 中对气球目标检测的可视化展示，我们可以直观地看到模型在实际应用中的表现。

6 总结与展望

本文聚焦于基于 Transformer 的目标检测方法 DETR，对其进行了详细的复现工作。在实验过程中，我们发现当使用单张显卡训练时，通过调整学习率能够达到与原始代码相似的性能水平。此外，我们将 VOC 数据集转换为 COCO 格式，以此验证了模型的通用性。

在原有代码的基础上，我们添加了中断控制功能，确保可以保存训练进度，并将这些成果上传至 HuggingFace 仓库，方便后续的研究和版本管理。为了增强模型对不同任务的适应

性，我们采用了多项优化策略：利用 StepLR 学习率调度器动态调整学习率，以促进更有效的收敛；启用混合精度训练，加速训练并降低内存消耗；还尝试了提示词微调，目前效果未达预期，但其潜力值得进一步挖掘。然而，尽管提示词微调理论上能够增强模型的灵活性和鲁棒性，但在实际应用中，我们观察到其效果并不如预期般显著，这可能与提示词的设计、数据预处理方式或学习率调整有关。

工作主要集中在对 DETR 模型的复现和适度调整上，但实验结果表明，DETR 在多种数据集上表现出良好的性能。然而，仍然存在进一步优化的空间，例如提升小目标检测的效果和加快训练速度。未来的工作可以探索多任务学习策略以及更高效的 Transformer 架构，以期拓展应用场景并提高性能表现。此外，针对提示词微调效果不佳的问题，可以进一步研究优化提示词设计、调整微调参数设置以及改进数据预处理流程，以寻求更好的微调效果。

参考文献

[1] Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

[4] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[5] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[6] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018.

[7] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

[8] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[9] T Lin. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[10] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018.

[11] Luis Pineda, Amaia Salvador, Michal Drozdzal, and Adriana Romero. Elucidating image-to-set prediction: An analysis of models, losses and datasets. *CoRR*, abs/1904.05709, 2019.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[13] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[14] S Hamid Rezatofighi, Vijay Kumar Bg, Anton Milan, Ehsan Abbasnejad, Anthony Dick, and Ian Reid. Deepsetnet: Predicting sets with deep neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5257–5266. IEEE, 2017.

[15] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[16] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[17] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[18] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. *CoRR*, abs/1912.02424, 2019.

[19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019.