DeepL
订阅DeepL Pro以翻译大型文件。
欲了解更多信息，请访问www.DeepL.com/pro。

Check for updates

利用变压器进行端到端物体检测

Nicolas Carion[1,2](✉), Francisco Massa[2], Gabriel Synnaeve[2], Nicolas Usunier[2], Alexander Kirillov[2], and Sergey Zagoruyko[2]

[1]巴黎道芬大学，法国巴黎
{alcinos,fmassa,gab,usunier,akirillov}@fb.com
[2]Facebook AI，美国门洛帕克
szagoruyko@fb.com

摘要我们提出了一种新方法，将物体检测视为一个直接的集合预测问题。我们的方法简化了检测流水线，有效地消除了对许多手工设计组件的需求，如非最大抑制程序或锚点生成，这些组件明确地编码了我们对任务的先验知识。新框架被称为 DEtection TRansformer 或 DETR，其主要成分是基于集合的全局损失（通过两端匹配强制进行唯一预测）和变换器编码器-解码器架构。DETR 给定了一小组固定的已学对象查询，通过推理对象之间的关系和全局图像上下文，直接得出并行的最终预测结果。与许多其他现代检测器不同，新模型概念简单，不需要专门的库。在极具挑战性的 COCO 物体检测数据集上，DETR 的准确度和运行时间性能与成熟且高度优化的 Faster R-CNN 基准相当。此外，DETR 可以很容易地通用于以统一的方式进行全视角分割。我们的研究表明，DETR 的性能明显优于竞争基线。训练代码和预训练模型见 https://github.com/facebookresearch/detr。

1 导言

物体检测的目标是为每个感兴趣的物体预测一组边界框和类别标签。现代检测器通过在大量建议集 [5，36]、锚点 [22] 或窗口中心 [45，52] 上定义代用回归和分类问题，以间接方式完成这组预测任务。它们的性能受到后处理步骤、锚集的设计以及将目标框分配给锚的启发式方法的显著影响 [51]。

为了简化这些管道，我们提出了一种直接集预测方法，以绕过代理任务。这种方法

电子版**补充材料** 本章的网络版（https:// doi.org/10.1007/978-3-030-58452-8 13）包含补充材料，授权用户可查阅。

Oc Springer Nature Switzerland AG 2020
A.Vedaldi et al. (Eds.)：ECCV 2020, LNCS 12346, pp.
https://doi.org/10.1007/978-3-030-58452-8_13

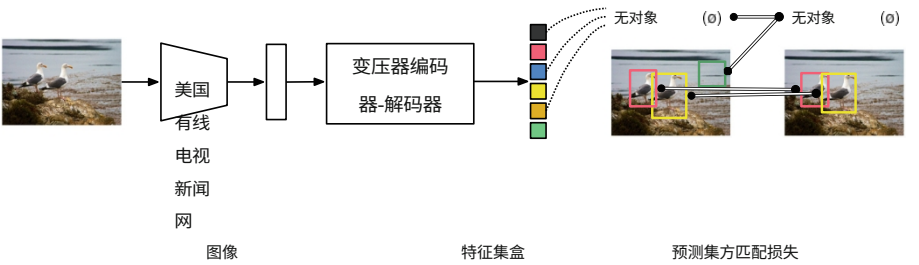

图 1.DETR 通过将普通 CNN 与变压器架构相结合，直接（并行）预测最终检测集。在训练过程中，双向匹配
将预测结果与地面实况框进行唯一分配。没有匹配的预测应产生 "无对象"（∅）类预测。

端到端理念已经在复杂的结构化预编译任务（如机器翻译或语音识别）中取得了重大进展，但在物体检测方面还没有：之前的尝试[4, 15, 38, 42]要么添加了其他形式的先验知识，要么没有在挑战性基准测试中证明与强大的基准线相比具有竞争力。本文旨在弥合这一差距。

我们将物体检测视为一个直接的集合预测问题，从而简化了训练管道。我们采用的编码器-解码器架构基于变换器 [46]，这是一种流行的序列预测架构。转换器的自我关注机制明确地模拟了序列中元素之间的所有成对交互，这使得这些架构特别适用于集合预测的特定约束，例如删除重复预测。我们的 DEtection TRansformer（DETR，见图 1）一次性预测所有对象，并使用集合损失函数进行端到端训练，在预测对象和地面实况对象之间执行双匹配。DETR 通过放弃多个编码先验知识（如空间锚点或非最大抑制）的人工设计组件来简化检测管道。与现有的大多数检测方法不同，DETR 不需要任何定制层，因此可以在任何包含标准检测方法的框架中轻松复制。
ResNet [14] 和 Transformer [46] 类。

与之前大多数关于直接集预测的研究相比，DETR 的主要特点是将双匹配损失和变换器与（非自回归）并行解码结合起来 [7、9、11、28]。相比之下，之前的工作主要集中在使用 RNN 的自回归解码 [29, 35, 40-42]。我们的匹配损失函数能将预测唯一地分配给地面实况对象，并且对预测对象的排列不变，因此我们可以并行地发射它们。

我们在最流行的物体检测数据集之一 COCO [23] 上对 DETR 进行了评估

，并与极具竞争力的 Faster R-CNN 基准 [36] 进行了对比。Faster R-CNN 经过了多次设计迭代，其性能自最初发布以来有了很大提高。我们的实验表明，我们的新模型取得了不相上下的性能。更准确地说，DETR 在处理大型物体时表现出了明显更好的性能，这可能得益于变换器的非本地计算。然而，它在小物体上的性能较低。我们希望未来的工作能像 FPN [21] 为 Faster R-CNN 所做的那样，在这方面有所改进。

DETR 的训练设置在多个方面与标准物体探测器不同。新模型需要超长的训练时间，并受益于 变压器中的辅助解码损耗。我们深入探讨了哪些组件对所展示的性能至关重要。

DETR 的设计理念很容易扩展到更复杂的任务中。在我们的实验中，我们发现在预先训练好的 DETR 的基础上训练的简单分割头在全景分割[18] 任务中的表现优于竞争基线，全景分割是一项具有挑战性的像素级识别任务，最近越来越受欢迎。

2 相关工作

我们的工作建立在以下几个领域的前期工作基础之上：用于集合预测的双匹配损失、基于变压器的编码器-解码器架构、并行解码和对象检测方法。

2.1 集合预测

目前还没有直接预测集合的典型深度学习模型。基本的集合预测任务是多标签分类（参见计算机视觉方面的参考文献，如[32, 39]），对于这种任务，基线方法（one-vs-rest）不适用于元素之间存在潜在结构（即近乎相同的方框）的检测等问题。这些任务的第一个难点是避免近似重复。目前大多数检测器都使用非最大抑制等后处理方法来解决这个问题，但直接集合预测则不需要后处理。它们需要全局推理方案来模拟所有预测元素之间的相互作用，以避免冗余。对于恒定大小的集合预测，密集的全连接网络[8]就足够了，但成本很高。一般的方法是使用自动回归序列模型，如递归神经网络[47]。在所有情况下，损失函数都应与预测值的排列保持不变。通常的解决方案是根据匈牙利算法[19]设计损失函数，在地面实况和预测之间找到两端匹配。这就强制实现了包络不变性，并保证每个目标元素都有唯一的匹配。我们采用的是双元匹配损失法。不过，与之前的大多数工作不同，我们不再使用自回归模型，而是使用具有并行解码功能的变压器，下文将对此进行介绍。

2.2 变压器和并行解码

Vaswani 等人[46]引入了转换器，作为机器翻译的一种基于注意力的新构建模块。注意力机制[2]是一种神经网络工作层，可汇总来自整个输入序列的信息。译者引入了自注意力层，它与非局部神经网络[48]类似，扫描序列中的每个元素，并通过聚合来自整个序列的信息进行更新。基于注意力的模型的主要优势之一是其全局计算和完美的记忆能力，这也是它的优势所在。

这使它们比 RNN 更适合处理长序列。在自然语言处理、语音处理和计算机视觉的许多问题中，变换器正在取代 RNN [7, 26, 30, 33, 44]。

变换器最早用于自动回归模型，沿用早期的序列到序列模型[43]，逐个生成输出标记。然而，令人望而却步的推理成本（与输出长度成正比，且难以批处理）导致了并行序列生成的发展，应用于音频 [28]、机器翻译 [9，11]、词表示学习 [7]，以及最近的语音识别 [6]。我们还将变换器和并行解码结合起来，以便在计算成本和执行集合预测所需的全局计算能力之间进行适当权衡。

2.3 物体检测

大多数现代物体检测方法都是根据一些初始猜测进行预测。两阶段检测器 [5, 36] 根据提议预测方框，而单阶段方法则根据锚点 [22] 或可能的物体中心网格 [45, 52] 进行预测。最近的研究[51]表明，这些系统的最终性能在很大程度上取决于设置这些初始猜测的具体方式。在我们的模型中，我们能够去除这种手工制作的过程，并通过与输入图像（而不是锚点）的绝对方框预测来直接预测检测集合，从而简化检测过程。

基于集合的损失。一些物体检测器 [8, 24, 34] 使用了双元匹配损失。然而，在这些早期的深度学习模型中，不同预测之间的关系仅用卷积层或全连接层建模，而人工设计的 NMS 后处理可以提高它们的性能。最近的检测器 [22, 36, 52]使用了地面实况和预测之间的非唯一赋值规则以及 NMS。

可学习的 NMS 方法[4, 15]和关系网络[16]明确地模拟了不同预测之间的关注关系。它们使用直接集合损失，不需要任何后处理步骤。然而，这些方法采用了额外的手工制作的上下文特征（如建议框坐标）来有效地模拟检测之间的关系，而我们则在寻找能减少模型中编码的先验知识的解决方案。

循环检测器。与我们的方法最接近的是用于物体检测 [42] 和实例分割 [29, 35, 40, 41] 的端到端集合预测。与我们的方法类似，它们使用基于 CNN 激活的编码器-解码器架构的双匹配损失来直接生成一组边界框。不过，这些方法仅在小型数据集上进行了评估，并未与现代基线进行对比。尤其是，这些方法基于自回归模型（更准确地说，是 RNN），因此无法利用最新的

并行解码转换器。

3 DETR 模式

探测中的直接集合预测有两个基本要素：(1) 集合预测损耗，它能使预测结果与地面实况进行唯一匹配

盒；(2) 预测（一次性）一组对象并对其关系进行建模的架构。图 2 详细描述了我们的架构。

3.1 物体检测集预测损失

DETR 一次通过解码器就能推导出固定大小的 N 组预测结果，其中 N 的设定值要远远大于图像中的典型物体数量。训练的主要困难之一是根据地面实况对预测对象（类别、位置、大小）进行评分。我们的损耗会在预测对象和地面实况对象之间产生最佳的两方匹配，然后优化特定对象（边界框）的损耗。

让我们用 y 表示物体的地面实况集，$\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ 表示的 N 个预测集合。假设 N 大于图像中物体的数量，我们将 y 也视为一个大小为 N 的集合，并填充∅（无物体）。为了找到这两个集合之间的双向匹配，我们要搜索成本最低的 N 元素 $\sigma \in \mathfrak{S}_N$ 的排列组合：

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

其中 $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ 是基本真相 y_i 与索引为 $\sigma(i)$的预测之间的成对*匹配成本*。根据之前的研究成果（*如* [42]），匈牙利算法可以高效地计算出这一最优分配。

匹配成本既要考虑类别预测，也要考虑预测方框与地面实况方框的相似度。地面实况的每个元素 i
可将目标类标签集视为 $y_i = (c_i, b_i)$，其中 c_i 是目标类标签（可能是∅），$b_i \in [0, 1]^4$ 是一个向量，定义了地面实况框的顶点坐标及其相对于图像大小的高度和宽度。对于
以 $\sigma(i)$为索引的预测，我们将类别 c_i 的概率定义为 $\hat{p}_{\sigma(i)}(c_i)$ 和
预测框为 $\hat{b}_{\sigma(i)}$。利用这些符号，我们将 $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ 定义为 $-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$.

这种寻找匹配的过程与现代检测器中用于将建议[36]或锚点[21]与地面实况对象相匹配的启发式分配规则的作用相同。主要区别在于，我们需要找到一对一的匹配，以实现无重复的直接集合预测。

第二步是计算损失函数，即上一步中所有匹配配对的*匈牙利损失*。我们对损失的定义与常见物体检测器的损失类似，即类预测的负对数似然与稍后定义的盒损失 $\mathcal{L}_{\text{box}}(\cdot, \cdot)$ 的线性组合：

$$L_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} -\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbf{1}_{\{c_i \neq \varnothing\}} \hat{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)), \quad (2)$$

其中 σ 是第一步 (1) 计算出的最优分配。在实践中，当 $c_i = \varnothing$ 时，我们会将对数概率项的权重降低 10 倍，以考虑以下因素

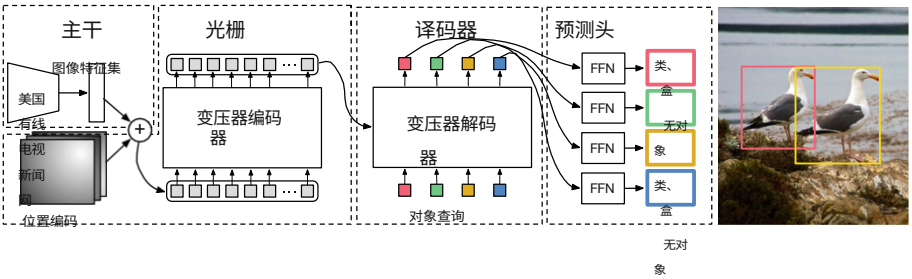

图 2.DETR 使用传统的 CNN 骨干来学习输入图像的二维表示。该模型将其扁平化，并辅以位置编码，然后将其传递给变换器编码器。然后，变换解码器将少量固定数量的已学位置嵌入（我们称之为*对象查询*）作为输入，并额外关注编码器的输出。我们将解码器的每个输出嵌入信息传递给共享前馈网络（FFN），该网络可预测检测结果（类别和边界框）或 "无对象 "类别。

类不平衡。这类似于 Faster R-CNN 训练程序通过子采样平衡正/负建议的方法 [36]。请注意，匹配
之间的成本并不取决于预测，这意味着
在这种情况下，成本就是一个常数。在匹配成本中，我们使用概率来计算。$\hat{p}_{\sigma(i)}(c_i)$ 而不是对数概率。这使得类预测项与 $L_{box}(\cdot,\cdot)$ 相称，我们观察到了更好的经验性能。

边框损失。匹配成本和匈牙利损失的第二部分是 $L_{box}(\cdot)$，它对边框进行评分。许多检测器会根据一些初始猜测对边框进行 Δ 预测，*与*此不同，我们直接对边框进行预测。虽然这种方法简化了实现过程，但却带来了损失相对缩放的问题。最常用的 l_1 损失对于小方框和大方框会有不同的比例，即使它们的相对误差相似。为了缓解这一问题，我们使用了 l_1 loss 和广义 IoU loss 的线性组合 [37]。
$L_{iou}(\cdot,\cdot)$ 是尺度不变的。总体而言，我们的箱体损失为 $L_{box}(b_i, \hat{b}_{\sigma(i)})$，定义为 $\lambda_{iou} L_{iou}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{L1} ||b_i - \hat{b}||_{\sigma(i)1}$ 其中 $\lambda_{iou}, \lambda_{L1} \in \mathrm{R}$ 是超参数。
这两项损失按批次内的对象数量进行归一化处理。

3.2 DETR 架构

DETR 的整体架构非常简单，如图 2 所示。它包含三个主要组件，我们将在下文进行介绍：用于提取紧凑特征表示的 CNN 主干网、编码器-解

码器转换器，以及进行最终检测预测的简单前馈网络（FFN）。与许多现代检测器不同，DETR 可以在任何深度学习框架中实现，只需几百行代码就能提供一个通用的 CNN 骨干网和一个变换器架构。在 PyTorch [31] 中，DETR 的推理代码只需不到 50 行即可实现。我们希望模拟我们方法的简便性将吸引新的研究人员加入检测领域。

骨干网。从初始图像 $x_{\text{img}} \in \mathrm{R}^{3\times H_0\times W_0}$（有 3 个颜色通道1）开始，传统的 CNN 主干网生成分辨率较低的激活图像
映射 $f \in \mathrm{R}^{C\times H\times W}$。我们使用的典型值是 $C = 2048$ 和 $H, W = H_0, \frac{W_0}{32}$。

变压器编码器。首先，1x1 卷积将高级激活图 f 的通道维度从 C 减小到更小的维度 d。

新特征图 $z_0 \in \mathrm{R}^{d\times H\times W}$。编码器希望输入一个序列，因此
我们将 z_0 的空间维度折叠为一维，得出 $d\times HW$
特征图。每个编码器层都采用标准架构，由多头自注意模块和前馈网络（FFN）组成。由于变换器架构是包络不变的，因此我们用固定位置编码[3, 30]对其进行补充，并将其添加到每个注意层的输入中。关于该架构的详细定义，我们将按照 [46] 中的描述在补充材料中进行说明。

变换器解码器。解码器沿用了变换器的标准架构，使用多头自我和编码器-解码器注意机制变换大小为 d 的 N 个嵌入。与原始转换器不同的是，我们的模型是在每个解码器层并行解码 N 个对象，而 Vaswani 等人[46]使用的是自回归模型，每次预测一个元素的输出序列。不熟悉这些概念的读者可参阅补充材料。由于解码器也是置换不变的，因此 N 个输入嵌入必须不同才能产生不同的结果。这些输入嵌入是学习到的位置编码，我们将其称为*对象查询，*与编码器类似，我们将其添加到每个注意力层的输入中。解码器将 N 个对象查询转化为输出嵌入。然后由一个前馈网络（将在下一小节中介绍）将它们*独立*解码为方框坐标和类别标签，从而得出 N 个最终预测结果。通过对这些嵌入的自我关注和编码器-解码器关注，该模型利用它们之间的配对关系对所有物体进行全局推理，同时还能将整个图像作为上下文。

预测前馈网络（FFN）。最终预测由一个具有 ReLU 激活函数和隐藏维数 d *的*三层感知器和一个线性投影层完成。FFN 预测输入图像中方框的归一化中心坐标、高度和宽度，线性层使用 softmax 函数预测类别标签。由于我们预测的是一个固定的
在 N 个边界框的大小集合中，N 通常远大于图像中感兴趣对象的实际数量，一个额外的特殊类标签 $\varnothing$ 被用来表示在一个槽中没有检测到任何对象。该类标签的作用是
其作用类似于标准物体检测方法中的 "背景 "类。

辅助解码损失。我们发现，在训练过程中，在解码器中使用辅助损失[1]是很有帮助的，尤其是可以帮助模型输出每一类对象的正确数量。每个解码器层的输出用以下公式归一化

[1]将 输入的图像分批处理，充分应用 0 填充，以确保所有图像的尺寸（H_0、w_0）与该批图像中最大的图像相同。

然后将共享层规范输入到共享预测头（分类和方框预测）。然后，我们会像往常一样将匈牙利损失用于监督。

4 实验

我们表明，在 COCO 的定量评估中，与 Faster R- CNN [36] 和 RetinaNet [22] 相比，DETR 取得了具有竞争力的结果。然后，我们对架构和损失进行了详细的消融研究，并提供了深入见解和定性结果。最后，为了证明 DETR 是一个通用模型，我们展示了全视角分割的结果，只对固定的 DETR 模型进行了少量扩展训练。

数据集。我们在 COCO 2017 检测和全景分割数据集[17, 23]上进行了实验，该数据集包含 118k 张训练图像和 5k 张验证图像。每张图像都标注了边界框和全景分割。每张图像平均有 7 个实例，在训练集中的单张图像中最多有 63 个实例，同一图像上的实例从大到小不等。如果没有指定，我们将 AP 报告为 bbox AP，即多个阈值的积分指标。为了与其他模型进行比较，我们报告了最后一个训练历时的验证 AP，在消融中，我们报告了最后 10 个历时的中位数。

技术细节。我们使用 AdamW [25] 训练 DETR，将初始转换器的学习率设为 10^{-4}，骨干网的学习率设为 10^{-5}，权重衰减设为 ^{10}-4。所有变换器权重均使用 Xavier init [10] 进行初始化，骨干网则使用来自 TORCHVISION 的经过 ImageNet 预训练的 ResNet 模型 [14] 和冻结的批规范层。我们报告了两种不同骨干网的结果：ResNet- 50 和 ResNet-101。相应的模型分别称为 DETR 和 DETR-R101。根据文献[20]，我们还在骨干层的最后一级增加了一个扩张，并在这一级的第一个卷积中去掉了一个跨距，从而提高了特征分辨率。相应的模型分别称为 DETR-DC5 和 DETR-DC5-R101（扩张的 C5 阶段）。这种修改将分辨率提高了 2 倍，从而提高了对小物体的处理性能，但代价是编码器的自我关注成本增加了 16 倍，导致计算成本总体增加了 2 倍。表 1 对这些模型、Faster R-CNN 和 RetinaNet 的 FLOP 进行了全面比较。

我们使用比例增强技术，调整输入图像的大小，使最短的

边长至少为 480 像素，最多为 800 像素，最长为 1333 像素 [49]。为了通过编码器的自我关注来帮助学习全局关系，我们还在训练过程中应用了随机裁剪增强，将性能提高了约 1 个 AP。具体来说，训练图像会以 0.5 的概率裁剪为随机矩形片段，然后再次调整大小为 800-1333。变换器的训练默认为 0.1。在推理时，一些插槽会预测空类。为了优化 AP，我们使用相应的置信度，用得分第二高的类别覆盖这些插槽的预测。这与过滤掉空槽相比，AP 提高了 2 分。其他训练超参数见附录。对于我们的消融

表 1.在 COCO 验证集上与带有 ResNet-50 和 ResNet-101 主干网的 RetinaNet 和 Faster R-CNN 的比较。上部显示的是 Detectron2 [49] 模型的结果，中部显示的是采用 GIoU [37]、随机作物训练时间增强和长 9 倍训练计划的模型的结果。DETR 模型的结果与经过大量调整的 Faster R-CNN 基线相当，AP_S 较低，但 AP_L 大幅提高。我们使用 torchscript 模型来测量 FLOPS 和 FPS。名称中不包含 R101 的结果对应 ResNet-50。

模型	GFLOPS/FPS	#params	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$
视网膜网	205/18	38M	38.7	58.0	41.5	23.3	42.3	50.3
更快的 RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
更快的 RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
更快的 RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
RetinaNet+	205/18	38M	41.1	60.4	43.7	25.6	44.8	53.6
更快的 RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
更快的 RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
更快的 RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

实验中，我们使用了 300 个历元的训练计划，200 个历元后学习率下降 10 倍，其中单个历元是对所有训练图像进行一次遍历。在 16 个 V100 GPU 上训练基线模型 300 个历元需要 3 天，每个 GPU 训练 4 幅图像（因此总批次大小为 64）。为了与 Faster R-CNN 进行更长时间的比较，我们训练了 500 个历元，在 400 个历元后学习率下降，从而将 AP 提高了 1.5 个百分点。

4.1 与更快的 R-CNN 和 RetinaNet 的比较

变压器通常使用亚当或阿达格拉德优化器进行训练，训练时间非常长，而且会出现辍学现象，DETR 也是如此。然而，更快的 R-CNN 是用 SGD 训练的，数据扩增极少，我们还没有发现 Adam 或 dropout 的成功应用。尽管存在这些差异，我们还是试图让我们的基线变得更强。为了

与 DETR 保持一致，我们在盒损失中添加了广义 IoU [37]，同样的随机作物扩增和长时间训练也能提高结果 [12]。结果见表 1。上半部分显示的是 Detectron2 Model Zoo [49]对 3x 计划训练的模型得出的结果。中间部分显示的是相同模型的结果（带 "+"号），但训练时使用了 9 倍进度表（109 个历时）和所述增强功能，总共增加了 1-2 个 AP。表 1 的最后一部分显示了多个 DETR 模型的结果。为了在参数数量上具有可比性，我们选择了一个具有 6 层变换器和 6 层解码器、宽度为 256、8 个注意力头的模型。与带有 FPN 的更快 R-CNN 一样，该模型

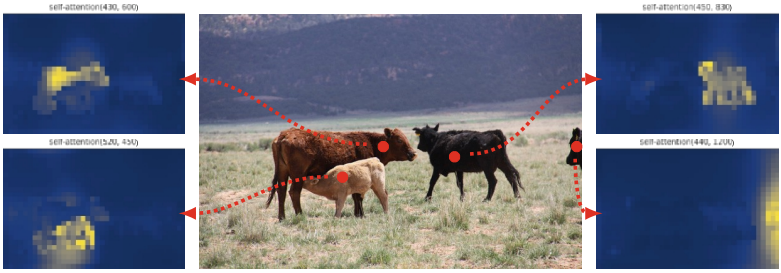

图 3.编码器对一组参考点的自我关注。编码器能够分离单个实例。使用基准 DETR 对验证图像进行预测。

有 4130 万个参数，其中 2350 万个在 ResNet-50 中，1780 万个在变压器中。尽管随着训练时间的延长，Faster R-CNN 和 DETR 仍有可能进一步提高，但我们可以得出结论：在参数数量相同的情况下，DETR 可以与 Faster R-CNN 竞争，在 COCO val 子集上实现 42 AP。DETR 实现这一目标的方法是提高 AP_L （+7.8），但请注意，该模型在 AP_S （-5.5）方面仍然落后。在参数数量相同、FLOP 数量相似的情况下，DETR-DC5 的 AP 更高，但在 AP_S 上仍明显落后。ResNet-101 主干网的结果也相当。

4.2 消融术

变压器解码器中的注意机制是模拟不同检测的特征表示之间关系的关键组件。在我们的消融分析中，我们探讨了架构的其他组件和损耗对最终性能的影响。在研究中，我们选择了基于 ResNet-50 的 DETR 模型，该模型有 6 层编码器和 6 层解码器，宽度为 256。该模型有 41.3M 个参数，在短调度和长调度上分别实现了 40.6 和 42.0 个 AP，运行速度为 28 FPS，与具有相同骨干网的 Faster R-CNN-FPN 相似。

编码器层数。我们通过改变编码器层数来评估全局图像级自我关注的重要性。在没有编码器层数的情况下，整体 AP 下降了 3.9 个点，其中大型物体的 AP 下降幅度更大，达到 6.0 个点。我们推测，通过使用全局场景推理，编码器对于区分物体非常重要。结果见附录。在图 3 中，我们将训练模型最后一个编码器层的注意力图可视化，重点关注图像中的几个点。编码器似乎已经在分离实例，这可能简化了解码器的对象提取和定位。

解码层数。我们在每个解码层之后应用辅助损失（见第 3.2 节），因此，预测 FFN 在设计上是为了从每个解码层的输出中预测对象而训练的。我们通过评估每个解码层预测的对象来分析每个解码层的重要性。

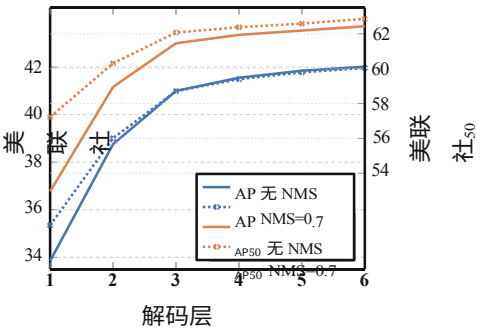

图 4.长时间表基线模型中每个解码器层之后的 AP 和 $_{AP50}$ 性能。DETR 在设计上不需要 NMS，这一点在本图中得到了验证。NMS 会降低最后几层的 AP 性能，因为 NMS 会移除 TP 预测，但会提高第一层的 AP 性能，因为 DETR 在第一层不具备移除双重预测的能力。

图 5.稀有类别的分布外泛化。尽管训练集中的长颈鹿数量不超过 13 只，但 DETR 对 24 只及以上的实例进行泛化并不困难。

解码阶段（图 4）。在每一层之后，AP 和 AP_{50} 都有所改善，在第一层和最后一层之间，AP 总共有 8.2/9.5 的显著改善。DETR 采用基于集合的损耗，在设计上不需要 NMS。为了验证这一点，我们使用默认参数 [49] 对每个解码器后的输出运行了标准 NMS 程序。NMS 提高了第一个解码器的预测性能。这是因为变换器的单个解码层无法计算输出元素之间的任何交叉相关性，因此容易对同一对象进行多次预测。在第二层和后续层中，激活的自我注意机制使模型能够抑制重复预测。我们观察到，随着深度的增加，NMS 带来的改进也在减少。在最后几层，NMS 会对 AP 造成伤害，因为它会错误地删除真正的正预测。

与编码器注意力可视化类似，我们在图 6 中也对解码器注意力进行了可视化，将每个预测对象的注意力图用不同颜色进行着色。我们观察到，解码器的注意力是相当局部的，这意味着它主要关注物体的四肢，如头部或腿部。我们假设，在编码器通过全局注意力分离出实例后，解码器只需注意物体的四肢即可提取出类别和物体的边界。

FFN 的重要性。转换器内的 FFN 可视为 1×1 卷积层，这使得编码器类似于注意力增强卷积网络 [3]。我们尝试将其完全移除，在变换器层中只留下注意力。通过将网络参数数量从 41.3M

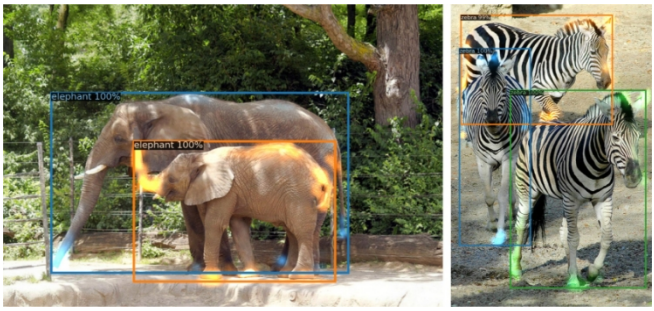

图 6.可视化解码器对每个预测对象的注意力（图像来自 COCO val set）。使用 DETR-DC5 模型进行预测。解码器通常会关注物体的四肢，例如腿部和头部。

因此，我们得出结论，FFN 对取得良好效果非常重要。

位置编码的重要性。我们的模型中有两种位置编码：空间位置编码和输出位置编码（对象查询）。我们尝试了固定编码和学习编码的各种组合，结果见附录。输出位置编码是必需的，而且无法移除，因此我们尝试在解码器输入时一次性传递这些编码，或者在每个解码器注意层将其添加到查询中。在第一个实验中，我们完全删除了空间位置编码，并在输入时传递输出位置编码，有趣的是，该模型仍然获得了超过 32 个 AP，比基线损失了 7.8 个 AP。然后，我们在输入时传递一次固定正弦空间位置编码和输出编码，就像最初的变换器一样[46]，结果发现这比在注意力中直接传递位置编码减少了 1.4 个 AP。将学习到的空间编码传递给注意力的结果类似。令人惊讶的是，我们发现在编码器中不传递任何空间编码只会导致 AP 下降 1.3 个百分点。当我们将编码传递给注意力时，它们会在所有层中共享，而输出编码（对象查询）始终是经过学习的。

鉴于这些缺陷，我们得出结论认为，变压器组件：编码器中的全局自我注意、FFN、多个解码器层和位置编码，都对最终的物体检测性能做出了重要贡献。

对未知数量的实例进行泛化。COCO 中的某些类别并不能很好地反映同一图像中同一类别的许多实例。例如，在训练集中就没有超过 13 只长颈鹿的图像。我们创建了一幅合成图像[2] 来验证 DETR 的泛化能力（见图 5）。我们的模型能够在图像上找到全部 24 只长颈鹿，这显然超出了分布范围。这

一实验证实了每个对象查询中并不存在很强的类别专一性。

[2]底图来源：https://www.piqsels.com/en/public-domain-photo-jzlwu。

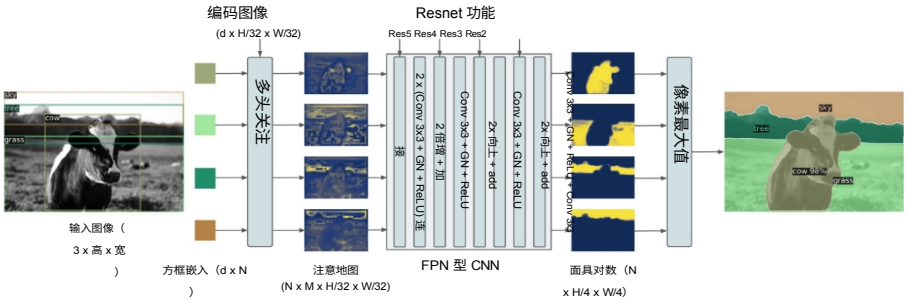

图 7.全景头示意图。为每个检测到的物体并行生成一个二元掩膜，然后使用像素级 argmax 合并掩膜。

4.3 DETR 用于全景分割

全景分割 [18] 最近引起了计算机视觉界的广泛关注。与 Faster R-CNN [36] 扩展到 Mask R-CNN [13]类似，DETR 也可以通过在解码器输出上添加一个掩码头来自然扩展。在本节中，我们将证明这种遮罩头可用于生成全景分割 [18]，以统一的方式处理东西和事物类别。我们在 COCO 数据集的全景注释上进行了实验，该数据集除了有 80 个事物类别外，还有 53 个事物类别。

我们使用相同的方法对 DETR 进行训练，以预测 COCO 上*东西*类周围的方框。预测方 框 是训练的必要条件，因为匈牙利语匹配是通过方框之间的距离来计算的。我们还添加了一个掩码头，为每个预测的方框预测二进制掩码，见图 7。它将变压器解码器对每个对象的输出作为输入，并在编码器的输出上计算该嵌入的多头（*M* 头）注意力分数，从而以较小的分辨率生成每个对象的 *M* 个注意力热图。为了进行最终预测并提高分辨率，我们使用了类似于 FPN 的架构。更多详情请参见增刊。掩码的最终分辨率为跨距 4，每个掩码都使用 DICE/F-1 loss [27] 和 Focal loss [22]进行独立监督。

掩码头可以联合训练，也可以分两步进行训练，即只训练 DETR 的方框，然后冻结所有权重，只训练掩码头 25 个历元。从实验结 果 来看，这两种方法得到的结果相似，由于后一种方法的计算量较小，因此我们采用后一种方法报告结果。为了预测最终的全景分割结果，我们只需对每个像素点的掩码得分进行 argmax 计算，并将相应的类别分配给得到的掩码。这

一过程可确保最终的掩码没有重叠因此，DETR 不需要启发式方法 [18] 来对齐不同的掩码。

训练细节我们按照边界框检测的方法训练 DETR、DETR-DC5 和 DETR-R101 模型，以预测 COCO 数据集中东西类周围的框。新的掩码头训练了 25 个历时（详见补充资料）。在推理过程中，我们首先过滤掉检测到的

表 2.在 COCO val 数据集上与最先进方法 UPSNet [50] 和 Panoptic FPN [17] 的比较 为了进行公平比较，我们对 PanopticFPN 进行了重新训练，使用了与 DETR 相同的数据增强，并采用了 18 倍的时间表。UPSNet 使用 1x 计划，UPSNet-M 是使用多尺度测试时间增强的版本。

模型	主干	PQ SQ RQ	PQth SQth RQth	PQst SQst RQst	美联社
PanopticFPN++	R50	42.4 79.3 51.6	49.2 82.4 58.8	32.3 74.8 40.6	37.7
UPSnet	R50	42.5 78.0 52.5	48.6 79.4 59.6	33.4 75.9 41.7	34.3
UPSnet-M	R50	43.0 79.1 52.8	48.9 79.7 59.7	34.1 78.2 42.3	34.3
PanopticFPN++	R101	44.1 79.5 53.3	**51.0 83.2** 60.6	33.6 74.0 42.1	**39.7**
DETR	R50	43.4 79.3 53.8	48.2 79.8 59.5	36.3 78.5 45.3	31.1
DETR-DC5	R50	44.6 79.8 55.0	49.4 80.5 60.6	37.3 **78.7** 46.5	31.9
DETR	R101	45.1 79.9 55.5	50.5 80.9 61.7	37.0 78.5 46.0	33.0
DETR-DC5	R101	**45.6 80.0 56.1**	50.9 80.9 **62.2**	**37.5** 78.6 **46.8**	33.1

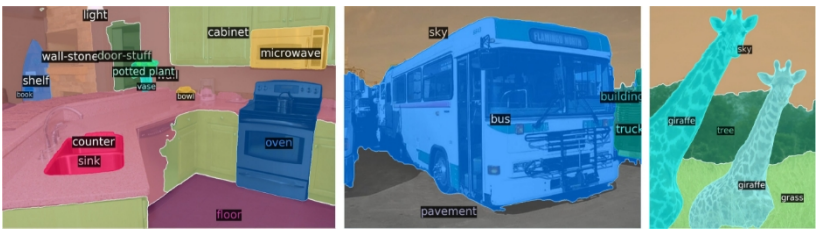

图 8.DETR-R101 生成的全景分割定性结果。DETR 以统一的方式为事物和物品生成对齐的掩码预测。

然后计算每个像素的 argmax，以确定每个像素属于哪个掩码。然后，我们将同一类别的不同掩码预测合并为一个，并过滤空掩码（小于 4 个像素）。

主要结果。定性结果如图 8 所示。在表 2 中，我们将统一的全景划分方法与几种不同处理事物和物品的成熟方法进行了比较。我们报告了全景质量（PQ）以及对事物（PQ^{th}）和物品（PQ^{st}）的细分。我们还报告了在进行任何全景后处理（在我们的案例中，是在提取像素级 argmax 之前）之前的掩码 AP（根据事物类别计算）。我们发现，DETR 优于 COCO-val 2017 上公布的结果，也优于我们强大的 PanopticFPN 基线（为进行公平比较，我们使用与 DETR 相同的数据增量进行训练）。结果细分显示，DETR 在东西类中尤其占优势，我们假设编码器注意力允许

的全局推理是这一结果的关键因素。对于事物类，尽管在掩码 AP 计算上与基线相比存在高达 8 mAP 的严重不足，但 DETR 仍获得了具有竞争力的 PQ^{th} 。我们还在 COCO 数据集的测试集上评估了我们的方法，并获得了 46 PQ。我们希望我们的方法能在未来的工作中激发人们探索全视角分割的完全统一模型。

5 结论

我们介绍了 DETR，这是一种新的物体检测系统设计，它基于直接集预测的转形器和双匹配损失。该方法在困难的 COCO 数据集上取得了与优化的 Faster R-CNN 基线相当的结果。DETR 简单易用，架构灵活，可轻松扩展到全视角分割，结果极具竞争力。此外，它在大型物体上的表现也明显更好，这可能归功于自我注意力对全局信息的处理。这种新的探测器设计也带来了新的挑战，特别是在小物体的训练、优化和性能方面。目前的探测器需要经过数年的改进才能解决类似问题、我们期待未来的工作能成功解决 DETR 的这些问题。

参考资料

1. Al-Rfou，R.、Choe，D.、Constant，N.、Guo，M.、Jones，L.：字符级语言建模与更深入的自我关注。In：AAAI 人工智能大会 (2019)
2. Bahdanau, D., Cho, K., Bengio, Y.：通过联合学习对齐和翻译的神经机器翻译。In：ICLR (2015)
3. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V..：注意力增强对话网络。In：ICCV (2019)
4. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-NMS-improving object detection with one line of code.In：ICCV (2017)
5. Cai, Z., Vasconcelos, N.: 级联 R-CNN：高质量对象检测和实例分割。PAMI (2019)
6. Chan, W., Saharia, C., Hinton, G., Norouzi, M., Jaitly, N.: Imputer: sequence modelling via imputation and dynamic programming. ArXiv:2002.08926 (2020)
7. Devlin，J.、Chang，M.W.、Lee，K.、Toutanova，K.：BERT：用于语言理解的深度双向变换器预训练。In：NAACL-HLT (2019)
8. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: 使用深度神经网络进行可扩展的物体检测。In：CVPR (2014)
9. Ghazvininejad, M., Levy, O., Liu, Y., Zettlemoyer, L.: Mask-predict: parallel decoding of conditional masked language models. ArXiv:1904.09324 (2019)
10. Glorot, X., Bengio, Y.：了解深度前馈神经网络的训练难度。In：AISTATS (2010)
11. Gu，J.、Bradbury，J.、Xiong，C.、Li，V.O.、Socher，R.：非自回归神经机器翻译。In：ICLR (2018)
12. He, K., Girshick, R., Doll´ar, P.：反思图像网络预培训。In：ICCV (2019)

13. He, K., Gkioxari, G., Doll´ar, P., Girshick, R.B.: Mask R-CNN.In：ICCV (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: 用于图像识别的深度残差学习。In：CVPR (2016)
15. Hosang, J.H., Benenson, R., Schiele, B.: Learning non-maximum suppression.In：CVPR (2017)
16. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.：用于物体检测的关系网络In：CVPR (2018)

17. Kirillov, A., Girshick, R., He, K., Doll´ar, P.：全景特征金字塔网络。In：CVPR (2019)
18. Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P. ： Panoptic segmentation.In：CVPR (2019)
19. 库恩（Kuhn, H.W.）：赋值问题的匈牙利法（1955 年
20. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.：全卷积实例感知语义分割。In：CVPR (2017)
21. Lin，T.Y.、Doll´ar，P.、Girshick，R.、He，K.、Hariharan，B.、Belongie，S.：用于物体检测的特征金字塔网络。In：CVPR (2017)
22. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Doll´ar, P.：密集物体检测的焦点丢失。In：ICCV (2017)
23. Lin, T.-Y., et al： Microsoft COCO: Common objects in context.In： Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014.LNCS, vol. 8693, pp.Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1 48
24. Liu, W., et al：SSD：单发多箱探测器。In：Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016.LNCS, vol. 9905, pp.Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0 2
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization.In：ICLR (2017)
26. Lu¨scher, C., et al：用于 LibriSpeech 的 RWTH ASR 系统：混合与关注--无数据增强。
27. Milletari, F., Navab, N., Ahmadi, S.A.: V-net：用于体积医学图像分割的全卷积神经网络。In：3DV (2016)
28. Oord, A ., et al： ArXiv:1711.10433 (2017)
29. Park, E., Berg, A.C.: Learning to decompose for object detection and instance segmentation. ArXiv:1511.06449 (2015)
30. Parmar, N., et al：图像转换器In：ICML (2018)
31. Paszke, A., et al：Pytorch：一个命令式的高性能深度学习库。In：NeurIPS (2019)
32. Pineda, L., Salvador, A., Drozdzal, M., Romero, A . : Elucidating image-to-set p r e - d i c t i o n : an analysis of models, losses and datasets. ArXiv:1904.05709 (2019)
33. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
34. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection.In：CVPR (2016)
35. Ren, M., Zemel, R.S.: 使用递归注意的端到端实例分割。In：CVPR (2017)
36. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks.PAMI **39** (2015)
37. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union.In：CVPR (2019)
38. Rezatofighi, S.H., et al：Deep perm-set net: Learn to predict sets with unknown permutation and cardinality using deep neural networks. arXiv:1805.00613 (2018)

39. Rezatofighi, S.H., et al：Deepsetnet：用深度神经网络预测集合。In：ICCV (2017)
40. Romera-Paredes, B., Torr, P.H.S..：递归实例分割。In：Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016.LNCS, vol. 9910, pp.Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4 19
41. Salvador, A., Bellver, M., Baradad, M., Marqu´es, F., Torres, J., Gir´o, X.：用于语义实例分割的循环神经网络。arXiv:1712.00617 (2017)

42. Stewart, R.J., Andriluka, M., Ng, A.Y.：拥挤场景中的端到端人员检测。In：CVPR (2015)
43. Sutskever, I., Vinyals, O., Le, Q.V.: 序列到序列的神经网络学习。In：NeurIPS (2014)
44. Synnaeve, G., et al：端到端 ASR：使用现代架构从监督学习到半监督学习。arXiv:1911.08460 (2019)
45. Tian，Z.、Shen，C.、Chen，H.、He，T.：FCOS：全卷积单级对象检测。In：ICCV (2019)
46. 瓦斯瓦尼，A.等人：注意力是你所需要的一切。In：NeurIPS (2017)
47. Vinyals, O., Bengio, S., Kudlur, M..：秩序问题：集合的序列对序列。In：ICLR (2016)
48. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks.In：CVPR (2018)
49. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2 (2019). https://github.com/facebookresearch/detectron2
50. Xiong, Y., et al：Upsnet：统一的泛光分割网络。In：CVPR (2019)
51. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.：通过自适应训练样本选择弥合基于锚和无锚检测之间的差距 arXiv:1912.02424 (2019)
52. Zhou, X., Wang, D., Kr¨ahenbu¨hl, P.：对象即点。arXiv:1904.07850 (2019)