计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(2):74−83 [doi: 10.15888/j.cnki.csa.009765] [CSTR: 32024.14.csa.009765]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于自适应 Token 池化与集合预测增强的目标检测①

刘　耀, 陈东方, 王晓峰

(武汉科技大学 计算机科学与技术学院, 武汉 430081)
通信作者: 刘　耀, E-mail: 210717351@qq.com

摘　要: 基于 Transformer 的目标检测算法往往存在着精度不足, 收敛速度慢的问题. 许多研究针对这些问题进行改进, 取得了一定的成果. 但是这些研究大都忽视了 Transformer 结构应用于目标检测领域时存在的两个不足之处. 首先, 自注意力运算结果缺乏多样性. 其次, 因集合预测难度大, 使得模型在匹配目标的过程中表现不稳定. 为了弥补上述缺陷, 首先设计了自适应 token 池化模块, 增加自注意力权重的多样性. 其次, 设计了一种基于粗预测的锚框定位模块, 并利用该模块为查询提供位置先验信息, 从而提高二分图匹配过程的稳定性. 最后, 设计了基于组的去噪任务, 通过训练模型对位于目标附近的正负查询进行区分, 从而提高模型进行集合预测的能力. 实验结果表明, 本文提出的改进算法在 COCO 数据集上取得了较好的训练结果. 与基线模型相比, 改进算法在检测精度与收敛速度上有较大提升.

关键词: 目标检测; query 初始化方式; 自注意力; 训练策略

引用格式: 刘耀,陈东方,王晓峰.基于自适应 Token 池化与集合预测增强的目标检测.计算机系统应用,2025,34(2):74–83. http://www.c-s-a.org.cn/1003-3254/9765.html

Object Detection Based on Adaptive Token Pooling and Enhanced Set Prediction

LIU Yao, CHEN Dong-Fang, WANG Xiao-Feng

(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: Transformer-based object detection algorithms often suffer from problems such as insufficient accuracy and slow convergence. Although many studies have proposed improvements to address these problems and have achieved certain outcomes, most of them overlook two key shortcomings when applying Transformer structure to the field of object detection. Firstly, self-attention computation results are not diversified. Secondly, due to the complexity of set prediction, the models are unstable during target matching. To overcome these deficiencies, this study proposes several enhancements. Firstly, an adaptive token pooling module is designed to increase self-attention weight diversity. Secondly, a rough-prediction-based anchor box localization module is introduced, which provides positional prior information for queries to enhance stability during bipartite matching. Lastly, a group-based denoising task is designed, which trains the model to distinguish between positive and negative queries near the target, thereby improving the model's ability to perform set prediction. Experimental results show that the proposed improved algorithm achieves better training results on the COCO dataset. Compared with the baseline model, the improved algorithm significantly outperforms in both detection accuracy and convergence speed.

Key words: object detection; query initialization mode; self-attention; training strategy

① 基金项目: 湖北省教育厅科学研究计划重点项目 (D20211106)
收稿时间: 2024-07-17; 修改时间: 2024-08-13; 采用时间: 2024-08-27; csa 在线出版时间: 2024-12-16
CNKI 网络首发时间: 2024-12-17

目标检测是计算机视觉中的一个基础任务, 主要内容是对于给定图片或视频, 将待检测目标在图像中的位置找出来并识别其类别. 随着设备计算能力的提高以及深度学习技术的发展, 基于深度学习的目标检测算法成为研究的热点, 广泛应用于人脸识别[1,2]、机器人导航[3,4]、智能视频监控[5,6]、交通场景检测[7,8]及航天航空等领域[9,10]. 自 Girshick 等人[11]提出 R-CNN 以来, 卷积神经网络 (convolutional neural network, CNN) 在目标检测领域一直占据着主导地位. CNN 可以充分地提取局部时空特征以实现较高精度地检测和识别. 但是卷积操作缺乏对图像全局信息的感知, 无法建模特征之间的依赖关系.

因此, 研究者们尝试将自然语言处理领域中的 Transformer[12]模型迁移到计算机视觉任务中. 相较于 CNN, Transformer 可以建模图像的全局依赖关系, 能够更加充分地利用上下文信息. 这种方法将目标检测视为一个直接的集合预测问题. 集合预测方法试图直接从输入数据中预测出一个不包含重复元素的目标集合. 这种方法的优势在于它可以避免复杂的后处理步骤, 简化模型的推理过程, 并提高模型对目标之间关系的理解.

基于上述观点, Liu 等人[13]提出一种将 Transformer 应用于目标检测任务的算法 DAB-DETR (dynamic anchor boxes are better query for DETR, DAB-DETR). 该算法对原有基于 Transformer 的目标检测算法进行了改进, 通过为查询引入动态更新的位置信息来提升模型的收敛速度, 并为查询赋予了可解释性.

但该工作忽视了基于 Transformer 的目标检测算法仍然存在的不足, 即自注意力运算结果缺乏多样性与二分图匹配过程不稳定. 这些缺陷进一步导致了模型在应用中收敛速度慢, 检测精度不足.

为了缓解上述问题, 本文提出了一种基于改进 DAB-DETR 的目标检测算法 TES-DETR. 改进工作如下.

为了增加自注意力运算结果的多样性, 设计了自适应 token 池化模块 (token-aware pooling, TAP), 从而增加自注意力运算结果的多样性, 提高 encoder 提取特征的质量与模型的鲁棒性.

为了对模型的集合预测能力进行增强, 首先设计了基于粗预测的查询初始化模块, 减小模型对位置信息的修正难度, 从而提高匈牙利匹配过程的稳定性. 随后设计了基于组的对比去噪任务 (group-wise contrastive denoising task, GCD-Task), 提高模型对目标附近的正负查询的分辨能力, 从而强化了模型的集合预测能力.

1 相关工作

1.1 DETR 算法的改进

Carion 等人[14]开创性地将 Transformer 模型应用到目标检测领域, 提出了 DETR (end-to-end object detection with Transformers) 算法. 然而该算法存在精度不足, 难以收敛的问题. 基于此, Zhang 等人[15]提出了基于对比去噪任务的端到端目标检测 (DETR with improved denoising anchor boxes, DINO) 算法. 其通过研究发现传统的去噪任务缺少判别负查询为无目标的能力. 基于这种观点, Zhang 等人提出在训练中为每一个真值框添加噪声, 让模型根据噪声的大小区分正负查询, 从而在后续的运算过程中对正查询进行重建, 并判别负查询为空.

Yao 等人[16]则提出了一种充分利用密集先验信息的端到端目标检测器 (improving end-to-end object detector with dense prior, Efficient DETR). 他们认为查询缺少位置先验信息造成了 DETR 收敛速度慢, 因此提出利用 encoder 输出的特征进行密集预测, 并利用密集预测为查询的初始化提供先验信息. 因此, decoder 的查询在前几层便获得了相对准确的位置信息, 减小了 decoder 对查询位置信息的迭代难度.

1.2 DAB-DETR 算法

Liu 等人通过实验论证了查询在模型中代表了什么, 并提出了基于改进 DETR 的目标检测算法 DAB-DETR. 具体来说, DAB-DETR 的第 1 处改进是将锚框信息融入查询的建模中去. 首先, 对锚框坐标进行独立的位置编码并在空间维度上进行连接. 随后, 利用 *MLP* 网络对编码信息进行处理. 具体过程如下:

$$\begin{aligned} P(A_q) &= P(x_q, y_q, w_q, h_q) \\ &= Cat(P(x_q), P(y_q), P(w_q), P(h_q)) \end{aligned} \tag{1}$$

$$Pos_q = MLP(P(A_q)) \tag{2}$$

其中, A_q 是锚框的位置信息, 使用四元组 (x_q, y_q, w_q, h_q) 来表示. $P(\cdot)$ 表示编码函数. $Cat(\cdot)$ 表示在维度上的连接操作. 此改进为查询提供了位置先验信息, 同时赋予了查询可解释性.

DAB-DETR 的第 2 处改进便是使用锚框的宽高信息来调制交叉注意力运算的关注区域. 具体过程如下:

$$ModulateAttn((x,\ y),(x_{\text{ref}},\ y_{\text{ref}}))\ =$$
$$\left(P(x)\cdot P(x_{\text{ref}})\frac{w_{q,\text{ref}}}{w_q}+P(y)\cdot P(y_{\text{ref}})\frac{h_{q,\text{ref}}}{h_q}\right)\Big/\sqrt{D} \quad (3)$$

其中, $w_{q,\text{ref}}/w_q$ 与 $h_{q,\text{ref}}/h_q$ 是调节关注区域大小的参数. 通过将锚框的宽高信息引入交叉注意力中, 从而改变交叉注意力关注区域的形状, 使得模型可以适应各种尺度与形状的目标.

最后, DAB-DETR 利用 *MLP* 网络预测锚框位置的偏移量, 从而对锚框的位置进行逐层地迭代修正. 随着层与层之间的迭代, decoder 得到的位置信息也越来越精确.

2 本文方法

2.1 网络框架

本文提出的基于自适应 token 池化与集合预测增强的端到端目标检测算法 TES-DETR (token-aware pooling and enhanced set prediction DETR, TES- DETR) 整体架构如图 1 所示.

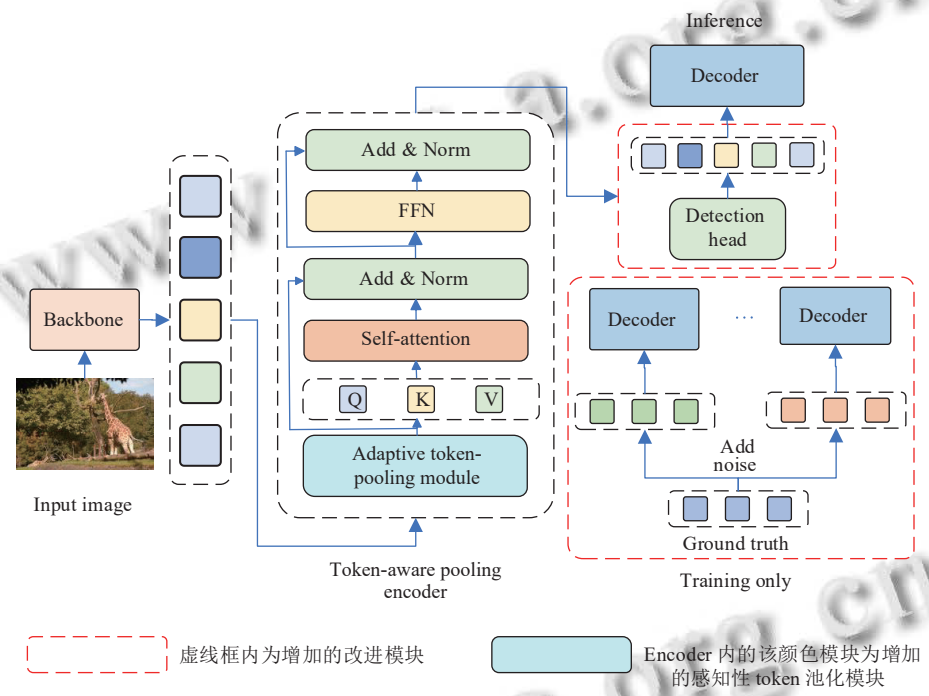


图 1 TES-DETR 网络结构图

TES-DETR 模型结构与 DAB-DETR 模型结构相比存在两处改进. 首先在 encoder 中加入了自适应 token 池化模块, 从而提升自注意力运算结果的多样性. 其次为了增强模型的集合预测能力, 本文在 encoder 之后设计了锚框定位模块, 并为 decoder 设计了组间查询串行输入的对比去噪任务.

图片首先输入到 backbone 中进行特征提取, 输出的特征信息经过处理便得到了 token 序列. 随后 encoder 接受输入的 token 序列, 并通过多个 encoder 层对输入序列进行逐层的编码, 用于捕捉输入序列中的上下文信息和特征表示.

TES-DETR 在每个 encoder 层中都加入了自适应 token 池化模块, 鼓励每个输入 token 从其局部邻域显式地聚合有用的信息, 以防该 token 本身不包含重要信息, 从而增加后续自注意力运算结果的多样性.

随后, 锚框定位模块接受 encoder 的输出并对图像中可能存在的目标进行预测. 该模块由基于空间的、基于任务的信息增强模块以及预测头组成. 其中, 信息增强模块可以对特征信息进行基于空间以及任务的强化, 从而提升检测头的预测效果. 最终检测头输出的预测将用来对查询的位置信息初始化.

在训练过程中, 本文提出的改进模型增加了基于组的对比去噪任务. 该任务可以提高模型对正负样本的区分能力, 加快收敛. 与传统的对比去噪任务不同, 本文设计了一种组间查询串行输入的方式. 这种方式减少了推理时查询的数量, 从而减小了同一位置存在

冗余查询的可能性.

2.2 自适应 token 池化模块

输入 encoder 中的 token 序列往往只有少数是包含重要信息的, 并且只有这部分少数的 token 可以得到较高的注意力分数. 这不但导致了自注意力运算结果缺少多样性, 也使 encoder 的抗干扰能力较差, 细微的噪声扰动就会造成自注意力运算结果的较大改变, 从而影响 decoder 抽取特征质量.

因此, 本文在自注意力运算之前加入自适应 token 池化模块, 缓解上述提到的问题. 具体结构如图 2.

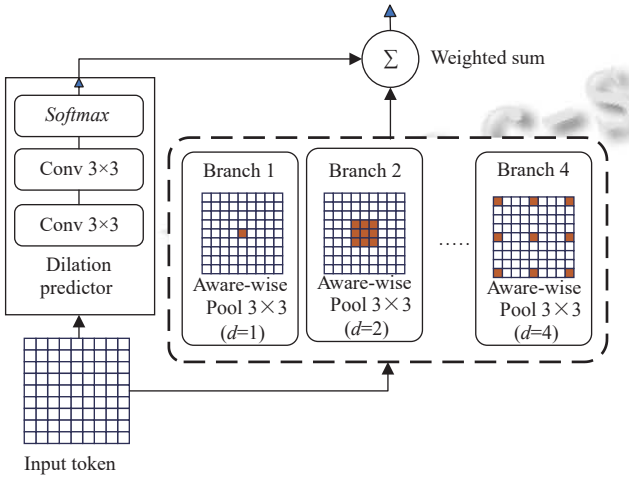


图 2 Token 池化模块

该模块首先对输入的 token 进行区域膨胀率不同的池化运算, 随后对这些运算结果进行加权求和, 从而自适应地调整池化区域. 具体运算过程如下所示:

$$T' = \lambda_1 Dpool_1(T) + \lambda_2 Dpool_2(T) + \cdots + \lambda_k Dpool_k(T) \tag{4}$$

$$\{\lambda_1, \lambda_2, \cdots, \lambda_k\} = Softmax(Conv(T)) \tag{5}$$

其中, T'为最终的运算结果, $Dpool_i(\cdot)$表示自适应池化运算, T表示输入的 token 序列. 集合$\{\lambda_i\}$是根据 token 信息预测的聚合权重, 预测过程是通过$Softmax(\cdot)$运算与$Conv(\cdot)$运算实现的. 与传统的区域固定的池化运算不同, 本文设计的自适应池化模块可以自适应地调整池化区域, 从而避免了相邻池化区域之间重叠过大, 进而导致输出 token 严重冗余的问题.

加入该模块之后, 各个 token 都或多或少地融合了部分重要 token 的信息. 这减少了对于重要 token 的过度依赖, 从而提高了模型的鲁棒性. 除此之外, 因该模块输出的 token 都或多或少的包含重要信息, 自注意力运算结果的多样性也大大提高了. 这间接地提高了 decoder 查询到的特征信息质量.

2.3 集合预测增强模块

DAB-DETR 在 DETR 算法的基础上引入了位置先验信息, 缓解了 DETR 算法收敛困难的问题. 但 DAB-DETR 在进行集合预测时仍然存在不稳定的问题, 需要训练 50 个轮次之后才能收敛. 造成这种结果的原因是查询对真值框的匹配是一个动态的、不稳定的过程. 对于一张图片来说, 在不同的轮次, 同一个查询通常会匹配到不同的真值框, 也就是说它的目标在频繁地切换. 这极大地增加了模型进行集合预测的难度.

为了增强集合预测能力, 本文设计了基于粗预测的查询初始化模块以及基于组的对比去噪任务. 模型首先利用 encoder 输出的特征信息进行预测, 输出存在目标可能性较大的位置信息. 输出的位置信息将作为查询的位置先验信息. 通过该模块的处理, decoder 将在前期的迭代过程中便获得相对精确的位置信息, 减小学习位置偏移量的难度, 从而增强模型进行集合预测的表现.

随后本文设计了对比去噪任务. 该模块首先对真值框添加噪声, 也就是对真值框的大小以及位置进行扰动, 生成正负样本. 随后进行去噪处理, 即逐步地对锚框的位置以及大小进行修正, 从而还原真值框的信息. 通过在训练中引入去噪任务有效地提升了二分图匹配的稳定性. 这是因为在执行去噪任务的过程中, 模型不仅需要修正真值框, 还需要对负样本进行抑制. 这极大程度上提高了模型对正负样本的辨别能力, 从而提高了二分图匹配过程的稳定性. 而这对于集合预测的效果是至关重要的.

2.3.1 基于粗预测的锚框定位模块

DAB-DETR 算法随机初始化 decoder 的查询. 这种方式虽然避免了对数据集中目标分布的拟合, 但是却增加了模型对锚框位置的修正难度, 影响模型的集合预测过程. 因此, 本文提出在 encoder 之后加入锚框定位模块, 并利用该模块输出的初步预测对查询的位置信息进行初始化. 该模块具体结构如图 3 所示.

该模块首先利用基于空间的运算模块与基于通道的运算模块对 encoder 输出的特征信息进行基于空间与任务的强化, 提高检测头输出预测的准确度. 随后利用 RPN head 对目标位置进行预测. 具体运算过程如下:

$$Q_i = \theta(RPN(E(f))) \tag{6}$$

$$e(F)=\pi_C(\pi_S(f)) \tag{7}$$

其中, $\theta(\cdot)$表示基于预测的锚框信息的 query 初始化模块, Q_i表示初始化的查询序列. $E(F)$表示特征信息强化模块, 该模块由基于空间的运算模块$\pi_S(\cdot)$与基于通道的运算模块$\pi_C(\cdot)$级联组成. $RPN(\cdot)$表示 RPN 检测头, 其利用强化后的特征信息输出锚框定位.

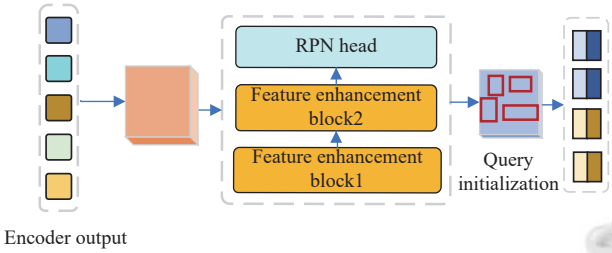


图 3 Query 初始化模块

前人的工作直接利用 encoder 输出的特征信息进行密集预测, 忽视了其并未经过进一步的提炼与深化, 这影响了检测头的预测效果. 因此本文提出在检测头前加入特征信息增强模块, 具体结构如图 4 所示.

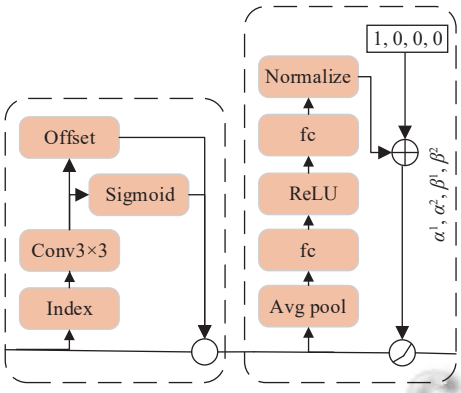


图 4 特征信息强化模块

该模块首先对输入的特征信息进行基于空间的增强, 随后进行基于任务的增强.

首先, 该模块对输入的特征信息进行空间维度上的稀疏采样, 通过学习位置偏移与不同位置的关注权重, 提取出感兴趣目标的完整特征. 运算细节如下:

$$\pi_S(f)=\sum_{k=1}^{K}\omega_k f(p_k+\Delta p_k;c)\cdot\Delta m_k \tag{8}$$

其中, $\pi_S(f)$代表基于空间的特征信息增强, f代表输入的特征信息, K代表稀疏采样个数. $p_k+\Delta p_k$代表着通过学习得到的位置偏移点. Δm_k表示根据各个位置特征学习得到的对不同偏移位置的关注权重.

随后, 基于通道的特征增强模块对特征信息进行下一步的处理. 首先对特征信息进行全局平均池化, 随后利用数个全连接层预测参数, 从而控制不同通道的开闭. 运算细节如下:

$$\pi_C(f)=\max(\alpha^1 f_C+\beta^1,\alpha^2 f_C+\beta^2) \tag{9}$$

其中, f_C表示对特征信息进行基于通道划分的结果. $[\alpha^1,\alpha^2,\beta^1,\beta^2]$则是通过学习得到的参数, 来控制不同通道的激活阈值. 该运算模块的引入可以动态的根据不同任务来控制不同特征图通道的开闭, 从而实现任务感知.

基于粗预测的锚框定位模块的具体步骤如算法 1 所示.

算法 1. 基于粗预测的锚框定位

输入: encoder 初步处理得到的 token 序列t_i.
输出: 包含了位置先验信息的查询序列Q_i.

1. 将 token 序列还原为特征图的形状$f=reshape(t_i)$
2. $\Delta p_k, \Delta m_k \leftarrow Conv(f)$
3. $f_1 \leftarrow \sum_{k=1}^{K}\omega_k f(p_k+\Delta p_k;c)\cdot\Delta m_k$
4. $f_C \leftarrow slice_{\text{channel}}(f')$
5. $f_2 \leftarrow \max(\alpha^1 f_C+\beta^1,\alpha^2 f_C+\beta^2)$
6. $anchor \leftarrow RPN(f_2)$
7. $Q_i \leftarrow q\_generation(anchor)$

综上所述, 本文提出的锚框定位模块可以提供更加精准的 anchor box 先验信息. 利用其作为查询的位置信息, 减少了交叉注意力过程中查询对无关区域的关注, 减小了模型对位置信息修正的难度, 最终提高了模型二分图匹配过程的稳定性.

2.3.2 基于组的对比去噪任务

前人提出将对比去噪任务引入 DETR 的训练过程中, 大大提高了模型对正负样本的辨别能力. 但传统的对比去噪任务将多组查询一起输入 decoder 中, 并分别计算去噪重建损失与匈牙利匹配损失. 这种方式因查询数量多, 导致了运算量较大. 与此同时, 推理过程中查询数量过多将导致目标位置附近存在冗余查询, 影响查询与真值框之间的匹配. 因此, 本文设计了基于组的对比去噪任务, 在运算过程中每组查询串行的输入 decoder 中, 依次进行预测任务以及去噪重建任务. 这种方式减少了推理过程中的查询数量, 提高了模型集合预测的能力. 该模块具体结构如图 5 所示.

该模块为训练过程中 decoder 部分的额外训练任务. 首先通过对真值框与标签添加噪声来构造正负查

询, 随后将其分组输入 decoder 中进行去噪重建任务. 与传统的对比去噪任务的执行方式不同, 本文提出的方法将多组正负查询串行地输入 decoder 中. 对于一张图片中的目标分布将进行 K 次回归. 具体过程如下所示:

$$\begin{cases} Decoder(X,Q_1)\to Q_1, Predictor(Q_1)\to Y_K \\ \vdots \\ Decoder(X,Q_K)\to Q_K, Predictor(Q_K)\to Y_K \end{cases} \tag{10}$$

其中, X 代表 encoder 输出的特征, K 表示正负查询的组的数量. Q_K 代表利用噪声生成的正负查询, $Predictor(\cdot)$ 表示利用查询输出目标预测 Y_K.

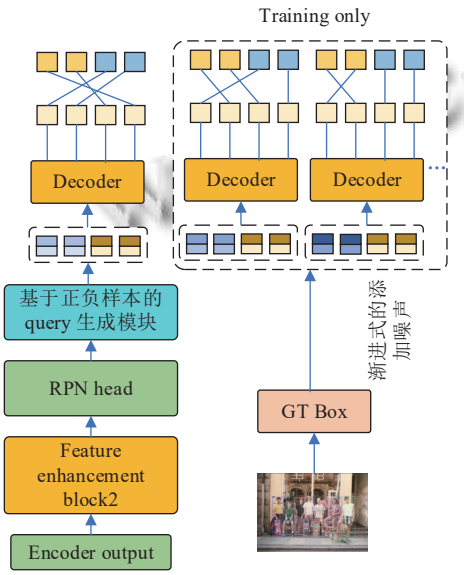


图 5 基于组的对比去噪 decoder 训练策略

同时, 本文对 query 的位置编码进行扰动. 首先利用参数控制真值框的宽高缩放, 随后再对真值框的位置添加噪声. 具体过程如下:

$$B' = \lambda_i B_{GT} + \lambda_j \tag{11}$$

其中, B' 表示经过扰动后的锚框, B_{GT} 表示真值框, 参数 λ_i 控制锚框大小的缩放, 参数 λ_j 控制锚框位置的扰动.

综上所述, 在基于组别的对比去噪任务训练中, decoder 展现出了卓越的正负样本区分能力, 特别是在接近真值框的查询中. 同时, 对于小位置扰动的查询, 该模型也展现出了良好的修正能力. 这些特性共同提升了模型在二分图匹配任务中的性能. 实验结果表明, 该模型展现出了卓越的检测效果.

2.4 损失函数

该模型首先利用匈牙利算法寻找使损失最小的一种匹配方式, 随后根据最佳匹配结果来计算本次预测的损失值. 除此之外, 该模型还需要计算去噪任务的重建损失. 模型整体损失函数的定义如下:

$$L = \lambda_1 \cdot L_{\text{Hungarian}}(y,\hat{y}) + \lambda_2 \cdot \frac{1}{K}\sum_{k=1}^{K} L_{re\_k} \tag{12}$$

其中, $L_{\text{Hungarian}}(y,\hat{y})$ 表示利用匈牙利算法得到最佳匹配之后计算得到的损失, 与 DETR 算法的损失函数保持一致. 而 L_{re_k} 表示基于组的重建损失. 模型分别计算 K 组去噪任务的重建损失并对其进行加和平均, 从而得到最终的重建损失. 每组的重建损失由 $L1$ 损失、GIoU 损失以及 focal 损失构成. 训练过程中的损失收敛曲线如图 6 所示.

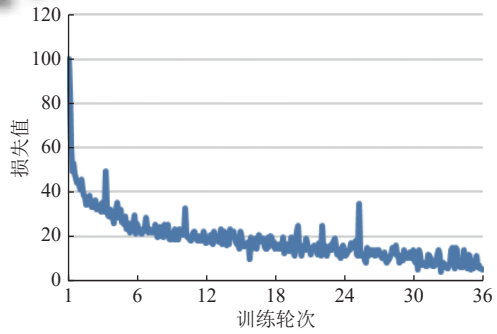


图 6 损失收敛曲线

3 实验结果与分析

3.1 数据集和评价指标

模型训练使用的数据集是 COCO 数据集. COCO (common objects in context)[17]是一个广泛用于目标检测、分割和关键点检测等任务的大规模数据集, 有超过 330k 张图像, 包含 150 万个目标, 80 个目标类别, 91 种材料类别, 包括人、动物、交通工具、家具等. 其中每张图像都有至少 5 个不同的注释. 这些注释包括 80 个不同类别的目标的边界框、实例分割掩码和关键点标注. 注释类型主要分为 3 种: 边界框标注 (每个目标都用一个矩形边界框来标识)、实例分割掩码 (每个目标都有一个像素级别的掩码, 用于准确地标识目标的轮廓)、关键点标注 (一些类别的目标还包含关键点标注, 用于识别特定身体部位的位置).

3.2 模型评价指标

在评价模型的目标检测性能时, 我们主要依据 *mAP* 指标来评估模型的检测精确度, 同时, 我们也参考 GFLOPs 指标来评价模型的计算复杂度. 另外, 为了

衡量模型对图片的处理速度, 我们还采用了 FPS 指标进行评估. 这些指标共同构成了我们全面评估模型性能的标准体系.

(1) 平均精度均值 (*mAP*): *mAP*0.5 是所有类别的 IoU 阈值在 0.5 时的平均检测精度. *mAP*0.5:0.95 是以步长为 0.05, 计算 IoU 阈值在 0.5–0.95 之间的所有 IoU 阈值下的平均检测精度. 公式如下:

$$mAP = \frac{1}{n}\sum_{i=1}^{n}\int_{0}^{1} P(R)\mathrm{d}R \tag{13}$$

(2) 浮点运算次数 (GFLOPs): 用于衡量模型在推理或训练过程中执行的浮点运算的总数量. 它是计算模型计算复杂度的指标之一, 用于评估模型的计算资源需求和效率.

(3) 模型参数量 (Params): 通常用于评估模型的复杂度和容量. 较多的参数量可能意味着模型具有更强的表示能力, 可以更好地适应训练数据, 但也可能增加过拟合的风险. 相反, 较少的参数量可能导致模型的表示能力不足, 难以捕捉数据中的复杂关系.

3.3 实验参数设置

本文的代码是基于 PyTorch 框架实现的, 版本为 2.0.0. Torchvision 的版本为 0.15.1, Cuda 版本为 11.7, 编译器是 Python 3.8. 服务器操作系统是 Ubuntu 20.04, 硬件主要采用了 NVIDIA GeForce RTX 3060 显卡, 显存为 12 GB.

特征提取的主干网络采用经过预训练的 ResNet-50[18]网络. Encoder 与 decoder 均是 6 层结构. 共有 4 级的多尺度的特征信息, 添加噪声的方式有两种, 一种是基于标签的, 另一种是基于锚框的. 为标签添加噪声的方式是随机将标签改编为另外的类别, 为锚框添加噪声的方式主要分为中心点漂移与框缩放这两种方法.

训练过程中一些超参数的设置如下. Batch-size 大小设置为 2, 单卡训练. 分支数 K 设置为 4. 因为批处理数量较小, 所以 batch norm type 设置为 FrozenBatchNorm. 学习率设置为 0.000 1, decoder 中 FFN 的激活函数为 ReLU, 标签噪声添加的比例设置为 0.5, 并使用指数滑动平均优化方式.

3.4 对比实验

为了验证本文提出的 TES-DETR 算法的具体效果, 本文在 COCO 数据集上对该算法的性能进行了实验. 同时, 为了证明该算法仅需要较少的训练就能得到较好的性能, 本文对仅训练了 12 个轮次的各个模型进行性能比较, 包括了 Anchor DETR、DN-DETR、Deformable DETR、DAB-DETR、Efficient-DETR 等基于 DETR 的主流目标检测算法. 实验结果如表 1 所示, 加粗数据为最优数值.

表 1 模型之间检测精度比较

Model	Epoch	*mAP* (%)	AP_{50} (%)	Params (M)	GFLOPs
Faster R-CNN[19]	12	37.9	58.8	40	207
DETR	12	15.5	29.4	41	225
Deformable DETR[20]	12	41.1	—	40	196
DAB-DETR	12	38.0	60.3	44	256
Dynamic DETR[21]	12	43.0	60.7	—	—
Efficient-DETR	12	39.1	59.4	54	289
Anchor DETR[22]	12	41.2	60.6	—	—
DN-Deformable-DETR	12	43.4	61.9	48	265
Dynamic head[23]	12	43.0	60.7	—	—
TES-DETR	12	**45.8**	**62.5**	48	273

从表 1 可以看出, Deformable DETR 与 DAB-DETR 无论是在 *mAP* 还是 AP_{50} 上的表现都远不如本文提出的算法. 本文提出的算法在 *mAP* 与 AP_{50} 上的表现分别比 Efficient-DETR 提升了 6.7%、3.1%, 比 DAB-DETR 提升了 7.8%、2.2%. 在参数量与浮点运算次数上, 本文提出的模型与主流模型相当, 仅是略有增加.

为了验证 TES-DETR 经过多轮训练之后的最佳检测精度表现如何, 本文在 COCO 数据集上对多个模型进行训练并比较. 如表 2 所示, 本文提出的模型经过 36 个轮次的训练便得到了最优的检测结果, 而目前主流的基于 DETR 的目标检测算法往往需要 50 个训练轮次甚至更多. 在性能上, 本文提出的算法在训练轮次更少的情况下, 在 *mAP* 与 AP_{50} 上的表现比 DAB-DETR 提升了 3.1%、2.9%, 这验证了本文提出的改进模块的有效性.

表 2 多轮次训练之后的精度比较 (%)

Model	Epoch	*mAP*	AP_{50}	AP_S	AP_M	AP_L
Faster R-CNN	108	42.0	62.4	20.5	45.8	61.1
DETR	500	43.3	63.1	22.5	47.3	61.1
Anchor DETR	50	45.1	65.7	27.4	50.1	60.9
Deformable DETR	50	46.2	65.2	28.8	49.2	61.7
DAB-DETR	50	46.6	66.0	30.1	50.4	62.5
DN-Deformable-DETR	50	48.6	67.4	31.0	52.0	63.7
Dynamic DETR	50	47.2	65.9	28.6	49.3	59.1
TES-DETR	**36**	**49.7**	**68.9**	**32.3**	**51.9**	**67.8**

3.5 消融实验

本文进行了消融实验来验证提出的各个模块的有效性, 结果如表 3 所示. 自适应池化模块的加入使

mAP 增长了 0.5%. 这是因为本文提出 TAP 模块鼓励每个 token 的局部邻域参与注意力机制, 以便可以自适应地考虑邻域中潜在重要 token 的信息, 从而在注意力图中获得更多具有高得分的列. 这提升了 encoder 提取特征的质量, 从而改善了模型整体表现. 检测头的加入使 *mAP* 增长了 0.2%. 这是因为该模块为 decoder 提供了更准确的 anchor box 先验信息, 减小了 decoder 对 query 信息进行迭代的难度. 基于组的对比去噪训练任务的加入使 *mAP* 增长了 2.3%, 这是因为对比去噪任务加强了 decoder 对正样本的回归能力以及负样本的抑制能力, 而基于组的结构增加了 decoder 对正样本回归的次数, 从而提高检测精度. 如图 7 所示, 本文提出的改进算法与基准模型相比在检测精度上有较大的提升, 明显地减少了小目标的漏检率.

综上所述, 综合 3 个方面改进后的模型 TES-DETR 与基准模型 DAB-DETR 对比, 在模型参数量和浮点运算数较少的提高同时, 其 *mAP* 和 AP_{50} 均有明显提升, 证明了改进后的模型能够更好地完成目标检测任务. 可视化实验结果对比如图 7 所示.

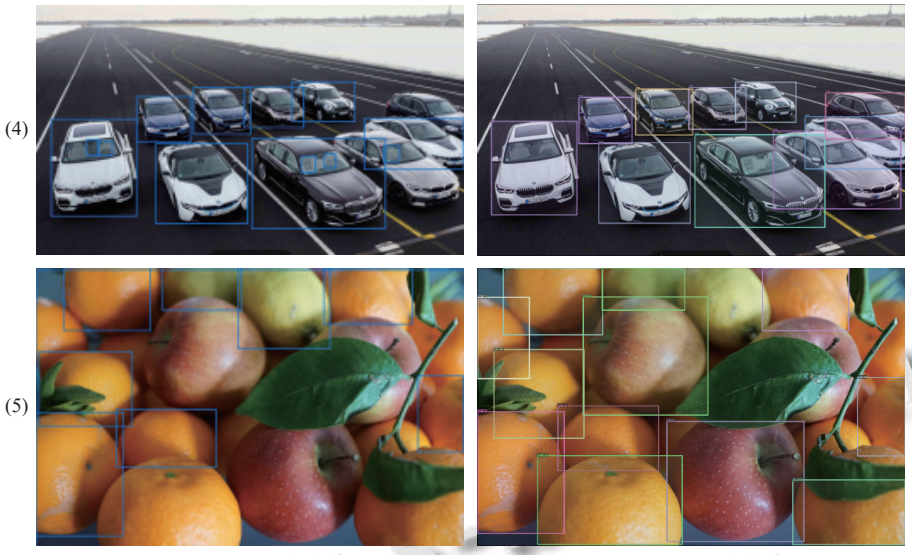
表 3 消融实验

TAP	Detect-head	Training-strategy	*mAP* (%)
—	—	—	46.6
√	—	—	47.1
—	√	—	46.8
—	—	—	—
—	—	√	48.9
√	√	√	49.7

4 结论与展望

本文提出基于 DAB-DETR 的改进目标检测算法. 为缓解 encoder 计算的注意力权重稀疏性与单调性, 本文在 encoder 的自注意力模块前加入自适应 token 池化模块, 鼓励每个 token 从其局部邻域显式地聚合有用的信息. 这使更多的 token 参与自注意力机制, 增加自注意力矩阵的多样性, 从而提升 encoder 提取特征的质量.

图 7 改进前后检测效果对比图



(a) DAB-DETR 可视化结果 (b) TES-DETR 可视化结果

图 7 改进前后检测效果对比图 (续)

与此同时, 本文设计了 query 初始化模块, 为输入 decoder 的 query 提供更加精确的位置先验信息, 从而减小二分图匹配任务的难度, 提升了收敛速度与检测精度. 通过训练时在 decoder 中引入基于组的真值框重建任务, 加强了 decoder 对于靠近真值框查询的预测能力与对负样本置信度的抑制能力. 这提升了模型在推理过程中的检测精度.

在 COCO 数据集上进行的实验证明了本文提出的改进模块的有效性. 通过对消融实验结果的分析, 则更进一步地阐释了各个模块在改进模型中是如何发挥作用的. 后续工作将研究该改进算法关于现实领域的具体应用问题.

参考文献

1 Liu YX, Yuan YX, Liu M. Ground-aware monocular 3D object detection for autonomous driving. IEEE Robotics and Automation Letters, 2021, 6(2): 919–926. [doi: 10.1109/LRA.2021.3052442]

2 Chen L, Wu PH, Chitta K, *et al*. End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. [doi: 10.1109/TPAMI.2024.3435937]

3 姬丽雯, 刘永华, 高菊玲, 等. 温室草莓采摘机器人设计与试验. 中国农机化学报, 2023, 44(1): 192–198. [doi: 10.13733/j.jcam.issn.2095-5553.2023.01.027]

4 肖德琴, 黄一桂, 熊悦淞, 等. 畜禽机器人技术研究进展与未来展望. 华南农业大学学报, 2024, 45(5): 624–634. [doi: 10.7671/j.issn.1001-411X.202404055]

5 杨飘. 基于深度学习的安防领域智能视频监控系统设计与实现. 信息记录材料, 2024, 25(11): 177–179. [doi: 10.16009/j.cnki.cn13-1295/tq.2024.11.022]

6 Kim HB, Choi N, Kwon HJ, *et al*. Surveillance system for real-time high-precision recognition of criminal faces from wild videos. IEEE Access, 2023, 11: 56066–56082. [doi: 10.1109/ACCESS.2023.3282451]

7 王义. 基于多功能路灯的智慧城市信息化建设研究. 智能建筑与智慧城市, 2020, (10): 13–14. [doi: 10.13655/j.cnki.ibci.2020.10.002]

8 Flores-Calero M, Astudillo CA, Guevara D, *et al*. Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review. Mathematics, 2024, 12(2): 297. [doi: 10.3390/math12020297]

9 刘佳铭. 基于 YOLOv5 的卫星遥感图像滑窗目标检测. 舰船电子工程, 2023, 43(1): 41–46. [doi: 10.3969/j.issn.1672-9730.2023.01.010]

10 闫钧华, 张琨, 施天俊, 等. 融合多层级特征的遥感图像地面弱小目标检测. 仪器仪表学报, 2022, 43(3): 221–229.

11 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014. 580–587.

12 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

13 Liu SL, Li F, Zhang H, *et al*. DAB-DETR: Dynamic anchor boxes are better queries for DETR. Proceedings of the 10th International Conference on Learning Representations. OpenReview.net, 2022.

14 Carion N, Massa F, Synnaeve G, *et al*. End-to-end object detection with Transformers. Proceedings of the 16th European Conference on Computer Vision. Glasgow: Springer, 2020. 213–229.

15 Zhang H, Li F, Liu SL, *et al*. DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection. Proceedings of the 11th International Conference on Learning Representations. Kigali: OpenReview.net, 2023.

16 Yao ZY, Ai JB, Li BX, *et al*. Efficient DETR: Improving end-to-end object detector with dense prior. arXiv:2104.01318, 2021.

17 Lin TY, Maire M, Belongie S, *et al*. Microsoft COCO: Common objects in context. Proceedings of the 13th European Conference on Computer Vision. Zurich: Springer, 2014. 740–755.

18 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

19 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 91–99.

20 Zhu XZ, Su WJ, Lu LW, *et al*. Deformable DETR: Deformable Transformers for end-to-end object detection. Proceedings of the 9th International Conference on Learning Representations. OpenReview.net, 2021.

21 Dai XY, Chen YP, Yang JW, *et al*. Dynamic DETR: End-to-end object detection with dynamic attention. Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021. 2968–2977.

22 Wang YM, Zhang XY, Yang T, *et al*. Anchor DETR: Query design for Transformer-based object detection. arXiv:2109.07107, 2021.

23 Dai XY, Chen YP, Xiao B, *et al*. Dynamic head: Unifying object detection heads with attentions. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 7369–7378.

(校对责编: 张重毅)