计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(10):106−114 [doi: 10.15888/j.cnki.csa.009639]

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

基于改进奖励机制的深度强化学习目标检测①

陈盈君, 武　月, 刘力铭

(长安大学 信息工程学院, 西安 710064)
通信作者: 陈盈君, E-mail: 1515222861@qq.com

摘　要: 为提高深度强化学习目标检测模型的检测精度和检测速度, 对传统模型进行改进. 针对特征提取不充分的问题, 使用融入通道注意力机制的 VGG16 特征提取模块作为强化学习的状态输入, 来更全面地捕获图像中的关键信息; 针对仅使用交并比作为奖励出现的评价不精准问题, 使用额外考虑了真实框与预测框中心点距离以及长宽比的改进奖励机制, 使奖励更加合理; 为加速训练过程的收敛并增强智能体对当前状态和动作评价的客观性, 使用 Dueling DQN 算法进行训练. 在 PASCAL VOC2007 和 PASCAL VOC2012 数据集上进行实验, 实验结果表明, 该检测模型仅需 4–10 个候选框即可检测到目标. 与 Caicedo-RL 相比, 准确率提高 9.8%, 最终预测框和真实框的平均交并比提高 5.6%.

关键词: 目标检测; 深度强化学习; VGG16; 注意力机制; 奖励机制; Dueling DQN

引用格式: 陈盈君,武月,刘力铭.基于改进奖励机制的深度强化学习目标检测.计算机系统应用,2024,33(10):106–114. http://www.c-s-a.org.cn/1003-3254/9639.html

Deep Reinforcement Learning for Object Detection Based on Improved Reward Mechanism

CHEN Ying-Jun, WU Yue, LIU Li-Ming

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

Abstract: To improve the detection accuracy and speed of deep reinforcement learning object detection models, modifications are made to traditional models. To address inadequate feature extraction, a VGG16 feature extraction module integrated with a channel attention mechanism is introduced as the state input for reinforcement learning, enabling a more comprehensive capture of key information in images. To address inaccurate evaluation caused by relying solely on the intersection over union as a reward, an improved reward mechanism that also considers the distance between the center points and the aspect ratio of the ground truth box and the predicted box is employed, making the reward more reasonable. To accelerate the convergence of the training process and enhance the objectivity of the agent's evaluation of current states and actions, the Dueling DQN algorithm is used for training. After conducting experiments on the PASCAL VOC2007 and PASCAL VOC2012 datasets, experimental results show that the detection model only needs 4–10 candidate boxes to detect the target. Compared with Caicedo-RL, the accuracy is improved by 9.8%, and the mean intersection over union between the predicted and ground truth boxes is increased by 5.6%.

Key words: object detection; deep reinforcement learning; VGG16; attention mechanism; reward mechanism; Dueling DQN

目标检测是计算机视觉领域中的研究热点, 一直备受关注. 其核心任务在于准确地识别并定位出图像或视频中的目标物体. 目标检测不仅能够为后续的视觉任务提供有力的支撑, 而且已经在各行各业中得到

① 收稿时间: 2024-03-28; 修改时间: 2024-05-06; 采用时间: 2024-05-23; csa 在线出版时间: 2024-08-28
CNKI 网络首发时间: 2024-08-29

了广泛应用. 现有的目标检测方法可以分为 3 大类: 传统方法、深度学习方法和深度强化学习方法.

传统方法依赖手工设计的特征提取器来进行目标检测. 采用滑动窗口策略, 通过在图像上滑动不同大小的窗口来选取潜在目标区域, 再使用如尺度不变特征变换 (scale-invariant feature transform, SIFT)[1]和方向梯度直方图 (histogram of oriented gradients, HOG)[2]等特定设计的提取器从这些区域中提取特征, 最后通过分类器进行分类和检测. 这类方法准确率低且难以适用于复杂场景.

基于深度学习的目标检测方法主要分为两阶段方法和单阶段方法. 两阶段方法首先筛选候选区域, 再进行精确检测. R-CNN[3]是最早的两阶段方法, 通过选择性搜索生成候选框, 并从中提取特征, 最后通过分类器和回归器得到检测结果. 然而, R-CNN 存在计算冗余且检测速度慢的问题. 为了改进这些问题, 后续的研究提出了 Fast R-CNN[4]、Faster R-CNN[5]、 Mask R-CNN[6]、Cascade R-CNN[7]等方法, 这些方法在不同方面提升了检测性能, 实现了更为高效的目标检测. 单阶段目标检测算法以 SSD[8]、YOLO[9]、RetinaNet[10]等为代表, 相较于两阶段方法, 其简化了检测流程, 直接将图像划分为网格并预测边界框, 从而大大提高了检测速度. 这些算法的发展为实际应用提供了更高效解决方案. 但这些方法往往参数量大且存在大量冗余候选框.

近年来, 一些基于深度强化学习的目标检测方法被提出. 这类方法将目标检测问题看作控制问题, 并建模为马尔可夫决策过程 (Markov decision process, MDP), 通过智能体与环境交互学习最优策略来实现对目标的定位与识别. Caicedo 等人[11]首次将深度强化学习用于目标定位任务, 通过智能体不断学习动作策略来搜索目标, 直至触发终止动作, 得到预测结果. 这种方法展示了深度强化学习在目标检测领域的潜力和有效性. 随后, Bellver 等人[12]提出了一种改进的深度强化学习目标检测方法, 采用分层策略. 该方法中, 智能体会先识别图像中的一个感兴趣区域, 然后逐步缩小这一区域, 进而在上一步选择的区域内继续寻找更小的目标区域, 最终形成层次化的检测结构. 这种分层策略有助于提高目标检测的精度和效率. 文献[13]提出的 Tree-RL 方法则采用自顶向下的搜索策略, 从整个图像开始逐步细化目标位置. 在每个搜索窗口, 算法递归地从动作空间中选取最佳动作, 生成新的窗口, 从而实现精准检测. Xu 等人[14]提出了一种名为 AHDet 的深度强化学习模型, 基于人类的视觉认知过程, 分为 AIM 和 HIT 两个步骤, 实现从粗略到精细的检测. 文献[15]侧重于让智能体学习纠正不准确边界框, 在粗定位的基础上实现边界框自动精细化. 文献[16]利用层次化的树型候选区域来减少评估数量, 并引入缩放和微调阶段、可变长宽比及多种奖励函数来提升检测效果. Reinforce-Net[17]模型中使用基于卷积神经网络的区域选择网络 (RS-net) 和边界框细化网络 (BBR-net) 来加强智能体的区域选择和细化能力.

为进一步提高检测效率, 本文提出一种基于改进奖励机制的深度强化学习目标检测模型. 同样将检测问题视为控制问题, 建立马尔可夫决策模型. 但使用融入通道注意力机制 (squeeze-and-excitation network, SENet)[18]的 VGG16 特征提取模块[19]作为强化学习的状态输入; 使用综合考虑了真实框与预测框中心点距离、长宽比差异以及交并比的改进奖励机制作为奖励; 使用位移、形变和缩放这 3 类定位动作; 利用 Dueling DQN[20]算法进行训练. 整个过程能够在使用较少候选框的情况下就精准定位到目标区域. 最后通过实验验证了本文模型的有效性.

1 基于改进奖励机制的深度强化学习目标检测模型

本节将目标检测过程建模为马尔可夫决策过程, 利用深度强化学习算法来让智能体做出最优决策. 基于改进奖励机制的深度强化学习目标检测模型如图 1 所示. 令定位框为智能体, 定位框每一时刻生成的预测框由 $b_t = \{x^t_{\min}, y^t_{\min}, x^t_{\max}, y^t_{\max}\}$表示, 其中$(x^t_{\min}, y^t_{\min})$为 t 时刻预测框的左上角坐标, $(x^t_{\max}, y^t_{\max})$为 t 时刻预测框的右下角坐标. 整体的检测流程为: 首先, 将图片输入模型中, 对图片进行预处理, 提取其特征向量作为初始状态. 接着, 定位框根据当前的状态选择一个动作, 执行该动作以调整预测框的大小或位置. 随后, 智能体进入新的状态, 同时系统会根据智能体的表现给予反馈值, 对动作进行奖励或惩罚, 从而引导智能体做出正确的决策. 循环往复, 智能体的目标是最大化累积的折扣奖励, 通过这个过程生成一系列的动作序列. 最终, 在满足特定条件时, 智能体会终止决策过程, 完成整个检测任务.

在本节, 对目标检测模型中的动作、状态、奖励以及训练策略进行详细介绍.

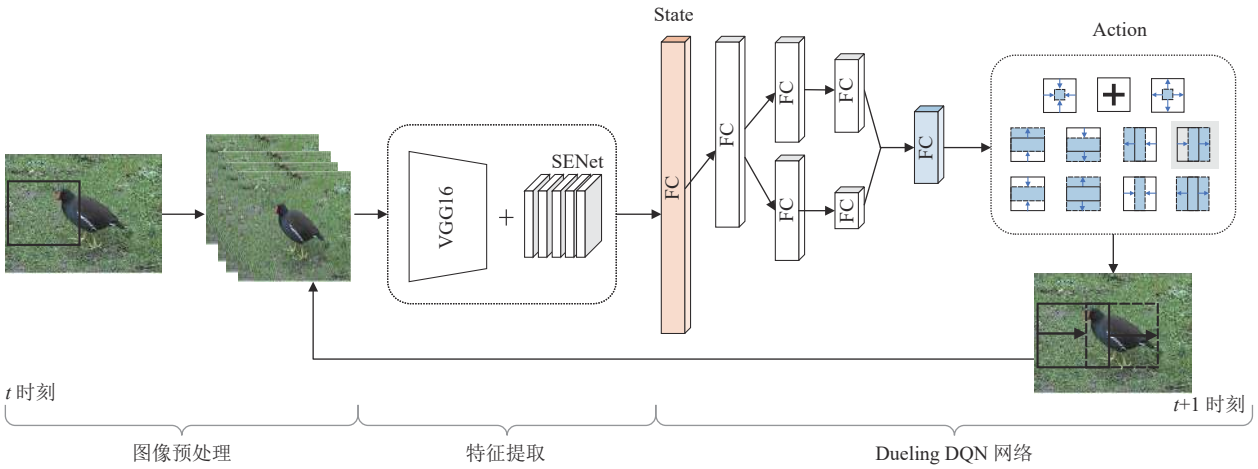

图 1 基于改进奖励机制的深度强化学习目标检测模型图

1.1 动作

在强化学习中, 动作是智能体为了达到目标而采取的行为, 通过执行动作, 智能体能够影响环境并获得相应的奖励或惩罚. 在本文模型中, 将动作定义为 $A=\{a_1,a_2,\cdots,a_t\}$, 其中 a_t 代表智能体在 t 时刻采取的动作. 对于动作的设计, 本模型设计了 10 种定位动作和 1 种终止动作. 定位动作用来改变预测框的位置和形状, 可细分为位移动作、形变动作和缩放动作, 如图 2 所示, 黑色实线框代表 t 时刻预测框的位置, 蓝色虚线框代表 t+1 时刻预测框的位置. 位移动作一共包含 4 种, 分别为向上平移、向下平移、向左平移和向右平移; 形变动作包含纵向压缩、纵向扩大、横向压缩和横向扩大 4 种; 缩放动作空间只包含等比例缩小和等比例放大.

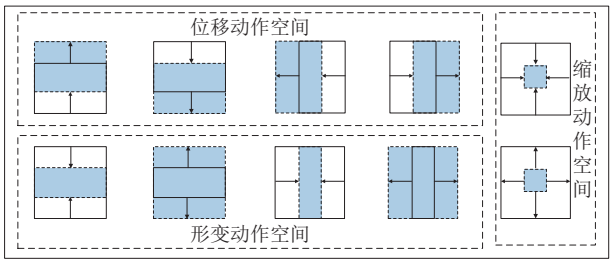

图 2 定位动作空间

定位动作可以对预测框进行离散的更改, 这些更改由特定的行动步长 Δx 和 Δy 来决定. 将变化系数定义为 μ, 取值在 0–1 之间, 代表更改幅度为当前预测框长或宽的 μ 倍. 例如, 执行向右平移动作时会将 Δx 添加到 $x^t_{\min}$ 和 $x^t_{\max}$ 上, 而 $y^t_{\min}$ 和 $y^t_{\max}$ 保持不变. 行动步长 Δx 和 Δy 可表示为:

$$\Delta x=\mu\cdot(x^t_{\max}-x^t_{\min}) \tag{1}$$

$$\Delta y=\mu\cdot(y^t_{\max}-y^t_{\min}) \tag{2}$$

此外, 在执行定位动作时, 若预测框因动作调整而超出图像边界, 模型会自动对其进行强制剪裁处理, 以确保预测框始终保持在图像的有效范围内. 以上动作空间的设定避免了冗余和重复动作的存在. 若变化系数的取值足够小, 定位框便可调整至任意形状和大小. 通过不同动作的组合, 定位框可以覆盖到图像中的任意一块区域.

终止动作用于终止序列过程. 当智能体执行终止动作时, 代表智能体认为当前预测框区域是目标所在区域, 即找到了物体, 会以黑色十字架形式进行标识. 此时, 智能体会结束当前搜索的序列, 并在初始位置重新启动该框, 以开始新一轮的搜索.

1.2 状态

在强化学习中, 状态是智能体在决策过程中的观察, 它包含了有关环境的必要信息, 以便智能体能够做出决策. 在本文模型中, 环境设为输入的图像, 将状态定义为 $S=\{s_1,s_2,\cdots,s_t\}$, s_t 代表智能体在 t 时刻观测到的信息. 本模型将经融合了 SENet[18]的 VGG16 特征提取模块[19]提取后的图像特征向量作为状态 S 输入.

具体来说, 将经过预处理后的图像经过 VGG16 网络进行特征提取以得到图像特征, VGG16 网络主要由 13 个卷积层、5 个池化层和 3 个全连接层组成. 其特点是结构相对简单, 没有太多的参数, 通过堆叠多个卷积层和池化层来提取图像的特征. VGG16 网络的卷积核大小为 3×3, 步长为 1; 最大池化操作的窗口尺寸是

2×2. VGG16 网络的结构图如图 3 所示. 将图像输入 VGG16 网络中, 得到经第 5 层池化层处理后产生的结果.

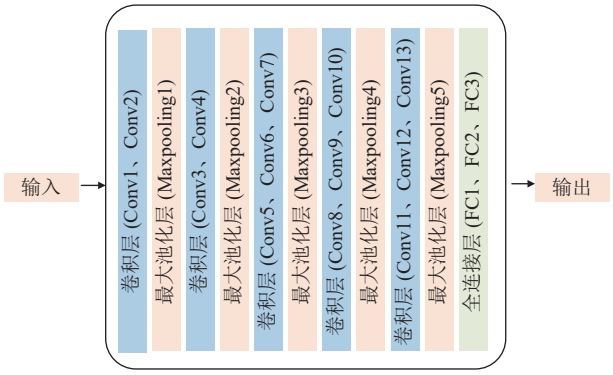

图 3 VGG16 模块

为显式地建模特征通道间的依赖关系, 提高网络模型对图像特征的提取能力. 在 VGG16 模型中加入 SENet 模块来让模型学习每个通道的重要程度, 进而强化重要的特征并抑制非重要的特征. 通道注意力机制模块的结构如图 4 所示. SENet 模块包含 3 个主要操作: Squeeze 操作、Excitation 操作和 Scale 操作. Squeeze 操作通过全局平均池化 (global pooling) 来聚合每个通道的全局信息, 得到一个表示通道全局特征的向量. Excitation 操作利用这个向量来预测每个通道的重要性. 首先经过第 1 个全连接层来进行降维操作, 减少计算量并捕获通道间的全局信息, 使用 ReLU 激活函数来引入非线性, 帮助模型学习更复杂的通道间关系. 接着经过第 2 个全连接层来进行升维操作, 将特征重新映射到与原始通道数相同的维度, 用 Sigmoid 激活函数将输出值限制在 0–1 之间, 生成归一化的通道权重. 随后, 通过 Scale 操作将权重加权到每个通道的特征上, 实现特征的重标定.

最终, 将经过 VGG16 模块和 SENet 模块处理后得到的特征图展平成一维向量作为目标检测模型的状态输入. 此外, 在模型中, 会取连续 4 帧图像作为输入, 使模型更全面地捕捉环境的动态性, 提高模型的感知能力和决策准确性.

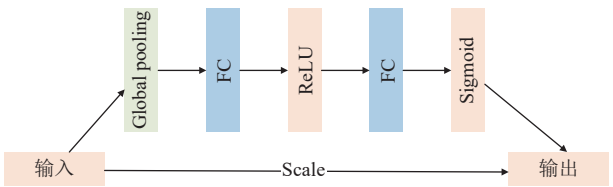

图 4 SENet 模块

1.3 奖励

在强化学习中, 奖励用于评估智能体在某个状态下采取行动的好坏, 并指导智能体逐步学习如何做出最优决策. 在本目标检测模型中, 将奖励定义为 $R=\{r_1,r_2,\cdots,r_t\}$, r_t 代表智能体在 t 时刻得到的奖励. 由于本任务中智能体的目的是通过执行一系列的动作来不断调整预测框的形状和位置, 找到最契合的预测框, 因此奖励函数应该引导智能体在每一步的行动中更加接近真实目标框.

交并比 (IoU) 作为衡量预测框与真实目标框相似度的指标, 是通过比较两者的交集区域与并集区域面积来实现的. 设 $g=\{x^g_{\min},y^g_{\min},x^g_{\max},y^g_{\max}\}$ 为真实目标框, 预测框 b_t 和真实框 g 之间的交并比可表示为:

$$IoU(t)=\frac{b_t\cap g}{b_t\cup g} \tag{3}$$

在 t 时刻, 当智能体选择动作 a_t 从状态 s_t 转换为 s_{t+1} 时, 智能体会得到一个奖励 r_t, 预测框的位置坐标变为 b_{t+1}, 预测框和真实目标框的交并比变为 $IoU(t+1)$, 将奖励 r_{IoU} 定义为:

$$r_{IoU}(t+1)=sign(IoU(t+1)-IoU(t)) \tag{4}$$

在此处的奖励中, 引入符号函数是为了更直观地量化奖励. 若 t+1 时刻的预测框优于 t 时刻的预测框, 便用 1 表示, 意为奖励; 若 t+1 时刻的预测框差于 t 时刻的预测框, 便用−1 表示, 意为惩罚; 若 t+1 时刻的预测框与 t 时刻的预测框相比, 并没有明显地接近或远离真实框, 便用 0 表示, 表示不奖励也不惩罚.

但仅使用 IoU 作为奖励, 会出现奖励不准确的情况, 如图 5 所示.

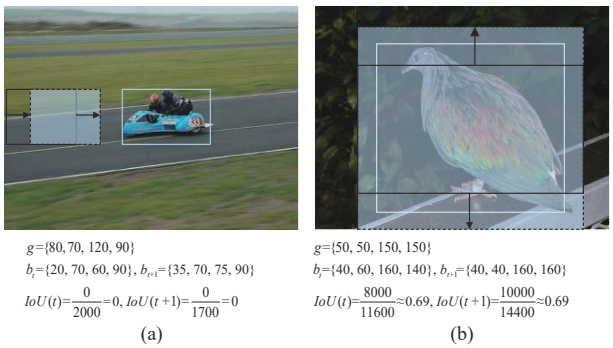

图 5 奖励不合理情况

图 5 中, 白色实线框代表真实框所在位置, 由 g 表示; 黑色实线框和蓝色虚线框分别代表 t 时刻和 t+1 时

刻预测框的位置, 由 b_t 和 b_{t+1} 表示; t 时刻和 t+1 时刻预测框与真实框的交并比分别由 $IoU(t)$ 和 $IoU(t+1)$ 表示.

在图 5(a) 中, t 时刻和 t+1 时刻预测框和真实框的交并比都是 0, 但明显可以看出, t+1 时刻预测框的位置要更接近真实框的位置; 在图 5(b) 中, 虽然 t 时刻和 t+1 时刻预测框和真实框的交并比都是 0.69, 但 t+1 时刻预测框的位置不仅完整框住了目标而且形状也更接近于真实框. 若只考虑 IoU 作为奖励的组成部分, 在这两类情况下, 都会给予智能体 0 的奖励, 但这显然是不合理的.

为了进一步完善这一评价体系, 在奖励机制中引入预测框和真实框中心点之间的距离 ($Dist$) 和高宽比 ($Ratio$) 作为补充信息. 距离的计算额外提供了预测框和真实框的位置信息, 高宽比的计算额外提供了预测框的形状信息, 而且都能避免在 IoU 相同但预测框具有差异的情况下, 智能体被给予不恰当奖励的情况, 进而确保智能体在评估过程中的客观性和准确性.

t 时刻的预测框 b_t 和真实框 g 之间的中心点距离定义为:

$$Dist(t)=\sqrt{\left(\frac{y^g_{\max}+y^g_{\min}}{2}-\frac{y^t_{\max}+y^t_{\min}}{2}\right)^2+\left(\frac{x^g_{\max}+x^g_{\min}}{2}-\frac{x^t_{\max}+x^t_{\min}}{2}\right)^2} \tag{5}$$

t 时刻的预测框 b_t 和真实框 g 之间的高宽比之差定义为:

$$Ratio(t)=\left|\frac{y^g_{\max}-y^g_{\min}}{x^g_{\max}-x^g_{\min}}-\frac{y^t_{\max}-y^t_{\min}}{x^t_{\max}-x^t_{\min}}\right| \tag{6}$$

当智能体从 t 时刻转变为 t+1 时刻, 距离变为 $Dist(t+1)$, 高宽比之差变为 $Ratio(t+1)$. 距离和高宽比引起的奖励 r_{Dist} 和 r_{Ratio} 仍采用符号函数的形式, 可表示为:

$$r_{Dist}(t+1)=-sign(Dist(t+1)-Dist(t)) \tag{7}$$

$$r_{Ratio}(t+1)=-sign(Ratio(t+1)-Ratio(t)) \tag{8}$$

至此, 便可构成一个更加准确的奖励机制. 具体来说, r_{IoU} 表示定位框在执行动作后与真实框的重叠程度是否变大; r_{Dist} 表示定位框选取动作后的预测框中心点是否更接近真实框中心点; r_{Ratio} 表示智能体选取动作后的形状是否更相似于真实框. 同时引入 α、β、χ 这 3 个系数来代表这 3 个指标对于奖励的重要程度. 总奖励 r 可表示为:

$$r(t)=\alpha\cdot r_{IoU}(t)+\beta\cdot r_{Dist}(t)+\chi\cdot r_{Ratio}(t) \tag{9}$$

使用改进后的奖励机制, 无论是图 5(a) 中的情况, 还是图 5(b) 中的情况, 系统都会给智能体奖励, 而不是既不奖励也不惩罚, 这是符合人们的认知的.

除此以外, 智能体选择终止动作时, 代表定位框认为已经检测到了目标, 此时会给智能体不同的奖励. 当预测框与真实框之间的 IoU 大于阈值δ时, 便视为定位正确. 终止动作的奖励表示为:

$$r(t_{\text{end}})=\begin{cases}+\eta, & \text{if } IoU(t_{\text{end}})\geqslant\delta\\ -\eta, & \text{else}\end{cases} \tag{10}$$

1.4 训练策略

在本模型, 使用 Dueling DQN 算法进行训练. 该算法有助于模型更好地理解和区分不同动作在当前状态下的潜在价值. Dueling DQN 算法沿用了 DQN 的基本框架, 使用深度神经网络来近似 Q 值函数, 将状态作为输入, 并输出每个动作的 Q 值. 然而, 与 DQN 不同的是, 在 Dueling DQN 中, 网络结构分为两部分: 共享层和特殊层. 共享层用于提取输入数据的特征, 特殊层则分为价值流和优势流. 价值流对应价值函数, 优势流对应优势函数. 价值函数输出一个实数, 用于表示当前状态的整体价值, 而优势函数则负责输出一个与动作数量等长的向量, 每个元素表示对应动作相对于其他动作的优势. 最后, 将价值函数和优势函数进行结合, 得到每个动作的最终 Q 值. 在训练过程中, Dueling DQN 使用与 DQN 相同的经验回放和目标网络技巧. 经验回放用于存储和重放过去的经验. 目标网络定期从主网络复制权重, 以稳定训练过程. 具体迭代过程如算法 1 所示.

算法 1. Dueling DQN 算法

初始化: 经验回放缓冲区 Ω, 状态-动作值函数Q的参数θ, 使用参数$\hat{\theta}\leftarrow\theta$初始化目标状态-动作值函数$\hat{Q}$.

for $episode$=1,2,··· do

  初始化环境并获取观测数据 o_1.

  初始化序列 $s_1=\{o_1\}$并对序列进行预处理 $\phi_1=\phi(s_1)$.

  for t=1,2,··· do

    通过概率 ϵ 选择一个随机动作 a_t, 否则选择动作 $a_t=\text{argmax}_a Q(\phi(s_t), a; \theta)$.

    执行动作 a_t 并获得观测数据 o_{t+1} 和奖励数据 r_t.

    设置 $s_{t+1}=\{s_t, a_t, o_{t+1}\}$并得到 $\phi_{t+1}=\phi(s_{t+1})$.

    存储状态转移数据 $(\phi_t, a_t, r_t, \phi_{t+1})$ 到 Ω 中.

从 Ω 中随机采样小批量数据 $(\phi_i, a_i, r_i, \phi_{i+1})$.

若 ϕ_{i+1} 不是终止状态, 设置

$Y_i=r_i+\gamma(V(\phi_{i+1};\hat{\theta}_v)+\max(A(\phi_{i+1},a_{i+1};\hat{\theta}_a))-\frac{1}{|A|}\sum_{a'}A(\phi_{i+1},a';\hat{\theta}_a))$.

否则, 设置 $Y_i=r_i$.

在 $(Y_i-Q(\phi_i, a_i; \theta))^2$ 上对θ执行梯度下降. 每隔 C 步对目标网络 $\hat{Q}$进行同步.

end for

end for

2 实验设置与评价指标

2.1 实验数据集

实验选用 PASCAL VOC2007 数据集和 PASCAL VOC2012 数据集. 这两个公开数据集一共包含猫、狗、飞机、船等 20 个类别. 标签采用 XML 文件的格式标注, 详细记录了每个目标对象在图像中的位置. 在训练过程中, 使用 PASCAL VOC2007 和 PASCAL VOC2012 两个数据集联合进行训练. 测试时, 在 PASCAL VOC2012 数据集中挑选 3 425 张图片作为测试集来评估模型的性能.

2.2 实验环境及参数配置

本文提出的基于改进奖励机制的深度强化学习目标检测模型采用 Python 作为开发语言, 在 TensorFlow 框架下进行实验. 具体配置信息如表 1 所示.

表 1 实验环境配置信息

名称	配置
操作系统	Ubuntu
处理器	Intel(R) Xeon(R) Silver 4214R CPU@2.40 GHz
显卡	RTX 3080 Ti (12 GB)
内存	90 GB
深度学习框架	TensorFlow 2.1
编程语言	Python 3.7

参数配置可分为马尔可夫建模过程参数配置和训练过程参数配置. 在马尔可夫建模过程中, 设置动作变化系数 μ 为 0.2; 将改进奖励机制中 $r_{IoU}(t)$、$r_{Dist}(t)$ 和 $r_{Ratio}(t)$ 的权重系数 α、β、χ 分别设为 0.7、0.2 和 0.1; 在终止动作的奖励中, 设置终止阈值 δ 为 0.5, 终止奖励η为 4. 在训练过程中, 设置模型与每张图像交互的回合数 *episode* 为 15, 批处理大小 *batch_size* 为 32, 折扣因子γ为 0.99; 经验回放缓冲区的初始值设为 500, 大小设为 500 000; ϵ-贪婪策略中的起始值设为 0.2, 终止值设为 1.0, 衰减步数设为 500; 将当前网络的权重参数复制给目标网络的间隔步数 C 设为 10 000 步.

2.3 评价指标

本文采用平均精度 (*mAP*) 和平均交并比 (*mIoU*) 作为评价指标来评估本文模型的性能. 平均交并比 (*mIoU*) 与交并比 (*IoU*) 有关, 交并比是预测框与真实框的交集面积和并集面积之比; 平均准确率 (*mAP*) 与准确率 (precision, *P*) 有关, 准确率指的是预测结果中预测正确的正样本个数占被模型预测为正样本的个数的比例, 评价指标的计算公式可表示为:

$$mIoU=\frac{\sum_{i=1}^{n}IoU^{(i)}}{n} \tag{11}$$

$$P=\frac{TP}{TP+FP} \tag{12}$$

$$mAP=\frac{\sum_{i=1}^{n}P^{(i)}}{n} \tag{13}$$

其中, *TP*、*FP* 表示正确框和误检框的数量, *n* 表示检测的总类别数.

3 实验结果分析

3.1 消融实验

为验证本文所提改进模型的有效性, 用 *mAP* 和 *mIoU* 作为评价指标, 在火车、牛、小轿车、羊、飞机、沙发、猫和椅子这 8 类上测试, 令 Caicedo-RL[11] 作为基线算法, 将文中所提部分分成 4 组 (即 VGG16 模块、SENet 模块、改进奖励机制 R′和 Dueling DQN 算法), 进行消融实验. 其中, “√”代表使用该策略, “×”代表不使用该策略. 实验结果如表 2 所示.

表 2 消融实验结果对比

模型	VGG16	SENet	R′	DDQN	*mAP* (%)	*mIoU* (%)
Caicedo-RL	×	×	×	×	47.6	55.1
1	√	×	×	×	47.0↓	55.3↑
2	×	√	×	×	48.8↑	56.4↑
3	√	√	×	×	50.9↑	59.1↑
4	×	×	√	×	54.8↑	58.8↑
5	×	×	×	√	51.5↑	57.2↑
6	√	√	√	√	57.4↑	60.7↑

在表 2 中可以看出, 如果仅使用 VGG16 模块特征提取后的结果作为状态输入, 准确率不会上升, 反而会下降 0.6%, 交并比仅提升 0.2%. 仅使用 SENet 模块, 准确率和交并比的提升也不明显. 但使用加入了 SENet 的 VGG16 特征提取模块后的结果作为状态输入, 准确

率有了 3.3% 的提升, 交并比有了 4% 的提升, 可以说明特征提取的越好, 最终得到的预测框就会与真实框越接近. 使用改进奖励机制后准确率有了 7.2% 的提升, 表明一个合理的奖励机制有利于智能体找到目标. 使用 Dueling DQN 算法进行训练, 无论是准确率还是交并比, 也都会有一定程度上的提升. 本文所提策略如果全部使用, 准确率会比 Caicedo-RL 模型提高 9.8%, 交并比会提高 5.6%.

3.2 对比实验

为进一步验证模型的性能, 将改进后的模型与 Caicedo-RL、Hierarchical-RL[12]和 Stefan-RL[16]这 3 个模型进行对比实验, 在测试集上进行测试, 测试集中一共包含 20 个类别. 同样使用 *mAP* 和 *mIoU* 作为模型的评价指标, 将火车、牛、小轿车、羊、飞机、沙发、猫和椅子这 8 类的结果进行展示. *mAP* 和 *mIoU* 的具体实验结果如表 3 和表 4 所示. 通过表 3 和表 4 可以看出, 改进后模型无论是在准确率还是交并比上, 都有一定程度的提升. *mAP* 达到 57.4%, 相比于 Caicedo-RL、Hierarchical-RL 和 Stefan-RL 模型, 分别提高 9.8%、24.3%、30.7%; *mIoU* 达到 60.7%, 比在 Caicedo-RL 模型上的测试结果提高 5.6%, 比在 Hierarchical-RL 模型的测试结果提高 19.6%, 比在 Stefan-RL 模型的测试结果提高 11.1%.

3.3 可视化结果

为了直观地展示智能体在决策过程中的表现, 将定位框在每一步选择动作后产生的预测框进行可视化. 图 6–图 8 展示了智能体是如何通过一步一步的动作来调整预测框的位置和形状, 并在合适的时机选择终止动作, 以达到最终的检测效果. 图 6 展示了智能体在检测大型目标时的过程, 以车和狗为例, 其在图片中的占比较大, 在检测定位车时仅用了 4 步, 检测定位狗时仅用了 2 步; 在图 7 中, 展示了智能体检测中型目标时的过程, 以检测牛为例, 牛在图片中的占比适中, 在检测牛时智能体采取了 7 次动作; 图 8 展示了智能体在检测小型目标时的过程, 以检测飞机为例, 该飞机在图片中占比较小, 智能体需要 12 步才可检测定位到目标. 本模型在检测定位小型目标时所需候选框个数要多于在检测中型目标和大型目标时所需的候选框个数.

表 3 对比实验 *mAP* 结果 (%)

模型	*mAP*	*P*							
		train	cow	car	sheep	aero	sofa	cat	chair
Caicedo-RL	47.6	56.4	42.2	59.1	49.5	54.2	38.3	52.8	28.5
Hierarchical-RL	33.1	45.6	21.2	23.6	20.3	44.8	36.2	55.1	18.3
Stefan-RL	26.7	45.4	17.6	16.2	18.6	43.1	15.1	49.6	7.4
本文	57.4	66.4	54.1	68.9	58.2	68.8	46.4	59.9	36.1

表 4 对比实验 *mIoU* 结果 (%)

模型	*mIoU*	*IoU*							
		train	cow	car	sheep	aero	sofa	cat	chair
Caicedo-RL	55.1	56.2	60.1	51.5	55.7	51.7	48.9	54.2	62.3
Hierarchical-RL	41.1	46.3	34.1	37.2	36.4	44.1	42.3	57.3	31.1
Stefan-RL	49.6	51.5	52.2	45.2	47.1	44.6	51.0	55.4	49.8
本文	60.7	61.5	64.5	52.0	57.3	56.7	61.4	67.0	64.9

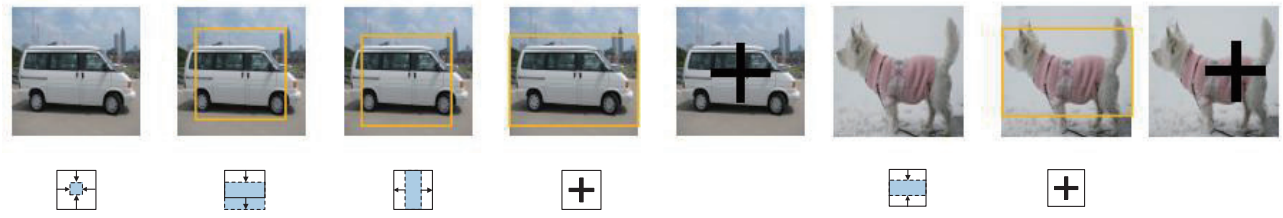

图 6 大目标检测过程可视化

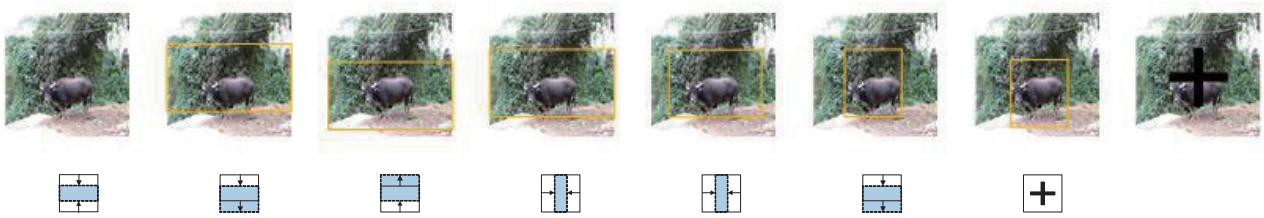

图 7 中目标检测过程可视化

图 9 中, 展示了智能体在检测定位物体时所需搜索步数的分布情况. 通过观察条形图, 可以发现, 大部分物体的检测定位过程所需的步数集中在第 4–10 步这一区间内. 一部分物体需要相对较多的搜索步骤, 即

第 29–37 步, 但这一比例相对较低. 这可能是由于这些目标物体占比过小或是因为这些目标物体所处的环境较为复杂, 导致智能体需要更多的步数来进行定位和搜索. 几乎没有物体需要超过 40 步才可以被定位到, 这一定程度上体现了智能体在检测定位过程中的高效性.

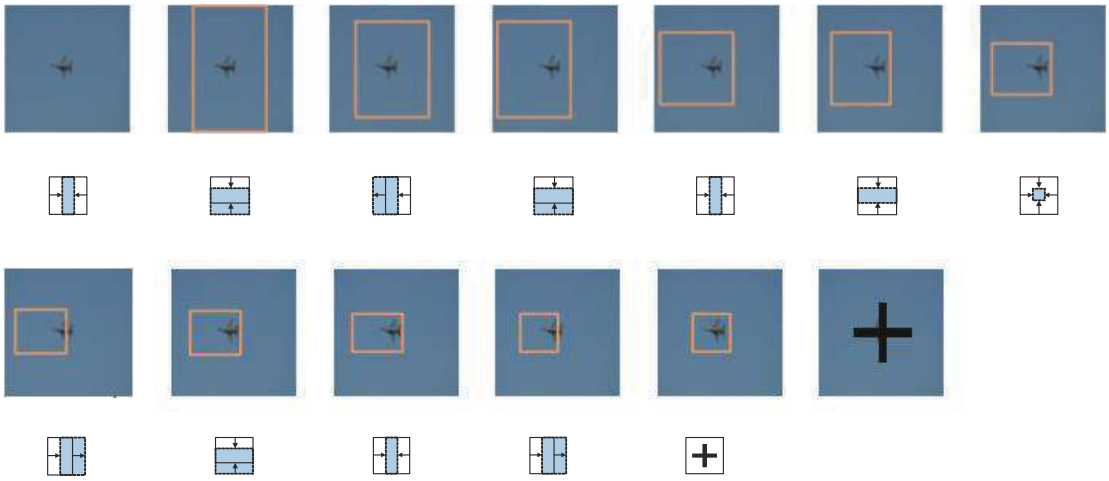

图 8 小目标检测过程可视化

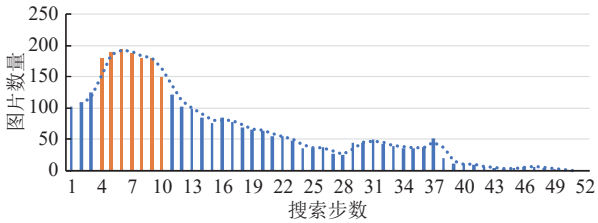

图 9 定位物体时所需搜索步数分布图

4 总结与展望

为解决深度强化学习目标检测模型中存在的检测精度低和速度慢的问题, 本文提出一种基于改进奖励机制的深度强化学习目标检测模型. 通过实验发现, 该模型在检测任务中有较好的表现, 可以在使用 4–10 个候选框后便检测定位到目标, 同时取得 57.4% 的准确度. 通过对强化学习状态部分的改进, 使模型对特征的提取更加充分, 通过对奖励的改进, 使模型对智能体每一步执行的动作有了更准确的评价, 通过对训练算法的改进, 使模型对状态和动作的认知更客观. 然而本文方法也存在一些不足, 如特征提取不充分、未充分考虑时间因素、对小目标检测效果差等, 仍需要对模型进行相关的改进. 针对特征提取不充分问题, 可选择更好的特征提取网络; 针对未考虑时间因素的问题, 可引入 GRU 等模块, 对小目标检测效果差问题, 可引入更精确的区域细化网络.

参考文献

1 Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110. [doi: 10.1023/b:visi.0000029664.99615.94]

2 Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego: IEEE, 2005. 886–893. [doi: 10.1109/CVPR.2005.177]

3 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. [doi: 10.1109/CVPR.2014.81]

4 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2015. 1440–1448. [doi: 10.1109/ICCV.2015.169]

5 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: 10.1109/TPAMI.2016.2577031]

6 He KM, Gkioxari G, Dollár P, *et al*. Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Venice: IEEE, 2017. 2980–2988. [doi: 10.1109/ICCV.2017.322]

7 Cai ZW, Vasconcelos N. Cascade R-CNN: Delving into high quality object detection. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 6154–6162. [doi: 10.1109/CVPR.2018.00644]

8 Yang J, Wang LQ. Feature fusion and enhancement for single shot multibox detector. Proceedings of the 2019 Chinese Automation Congress (CAC). Hangzhou: IEEE, 2019. 2766–2770. [doi: 10.1109/CAC48633.2019.8996582]

9 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 779–788. [doi: 10.1109/CVPR.2016.91]

10 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Venice: IEEE, 2017. 2999–3007. [doi: 10.1109/ICCV.2017.324]

11 Caicedo JC, Lazebnik S. Active object localization with deep reinforcement learning. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2015. 2488–2496. [doi: 10.1109/ICCV.2015.286]

12 Bellver M, Giró-i-Nieto X, Marqués F, *et al*. Hierarchical object detection with deep reinforcement learning. arXiv: 1611.03718, 2016.

13 Jie ZQ, Liang XD, Feng JS, *et al*. Tree-structured reinforcement learning for sequential object localization. Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 127–135.

14 Xu N, Huo CL, Zhang X, *et al*. AHDet: A dynamic coarse-to-fine gaze strategy for active object detection. Neurocomputing, 2022, 491: 522–532. [doi: 10.1016/j.neucom.2021.12.030]

15 Ayle M, Tekli J, El-Zini J, *et al*. Bar —A reinforcement learning agent for bounding-box automated refinement. Proceedings of the 34th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020. 2561–2568. [doi: 10.1609/aaai.v34i03.5639]

16 König J, Malberg S, Martens M, *et al*. Multi-stage reinforcement learning for object detection. Proceedings of the 2019 Computer Vision Conference. Las Vegas: Springer, 2019. 178–191. [doi: 10.1007/978-3-030-17795-9_13]

17 Zhou M, Wang RJ, Xie CJ, *et al*. ReinforceNet: A reinforcement learning embedded object detection framework with region selection network. Neurocomputing, 2021, 443: 369–379. [doi: 10.1016/j.neucom.2021.02.073]

18 Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7132–7141. [doi: 10.1109/CVPR.2018.00745]

19 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proceedings of the 3rd International Conference on Learning Representations. San Diego: ICLR, 2015.

20 Wang ZY, Schaul T, Hessel M, *et al*. Dueling network architectures for deep reinforcement learning. Proceedings of the 33rd International Conference on Machine Learning. New York: JMLR.org, 2016. 1995–2003.

(校对责编: 张重毅)