计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(1):297−303 [doi: 10.15888/j.cnki.csa.009381]

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

基于深度可分离卷积和交叉注意力的水面污染识别①

王　宁[1,2], 杨志斌[1,2]

[1](中国科学院 沈阳计算技术研究所, 沈阳 110168)
[2](中国科学院大学, 北京 100049)
通信作者: 杨志斌, E-mail: yangzhibin21@mails.ucas.ac.cn

摘　要: 水面污染严重影响水面景观和水体生态. 针对识别水面污染过程中水面场景复杂、小目标污染物特征难以提取等问题, 本文提出一种基于深度可分离卷积与交叉注意力算法模块 (deep-wise convolution and cross attention, DCCA). 使用深度可分离卷积降低模型的参数量和计算量, 使用交叉注意力建立不同尺度特征图之间的关系, 使模型更好地理解上下文信息并提高识别复杂场景和小目标的能力. 实验结果表明, 添加 DCCA 模块后平均精确率提升了 1.8%, 达到了 88.7%. 并使用较少的显存占用提高了水面污染的检测效果.

关键词: 深度可分离卷积; 交叉注意力; 污染识别; 目标检测; 卷积神经网络; 深度学习

引用格式: 王宁,杨志斌.基于深度可分离卷积和交叉注意力的水面污染识别.计算机系统应用,2024,33(1):297–303. http://www.c-s-a.org.cn/1003-3254/9381.html

Water Surface Pollution Recognition Based on Deep-wise Convolution and Cross Attention

WANG Ning[1,2], YANG Zhi-Bin[1,2]

[1](Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Water pollution seriously affects the water landscape and water ecology. In this study, a deep-wise convolution and cross attention (DCCA) algorithm module is proposed to address the issues of complex water surface scenes and difficulty in extracting features of small target pollutants in the process of identifying water surface pollution. The use of deep-wise convolution reduces the parameters and computational complexity of the model, and establishes relationships between feature maps at different scales using cross attention, enabling the model to better understand contextual information and improve its ability to recognize complex scenes and small targets. The experimental results show that the average accuracy has been improved by 1.8% after adding the DCCA module, reaching 88.7%. The detection effect of water surface pollution has been improved by using less memory occupation.

Key words: deep-wise convolution; cross attention; pollution recognition; object detection; convolutional neural network (CNN); deep learning

1　引言

随着经济的快速发展和工业化的不断推进, 人们生活水平得到提高, 同时对生态环境的污染也日益加剧. 特别是对于水环境的污染, 威胁了人类社会的可持续发展, 对人类健康构成了极大威胁.

水环境污染中, 水面污染是具有感官性, 易被观测

① 基金项目: 辽宁省应用基础研究计划 (2022JH2/101300126); 沈阳市中青年科技创新人才支持计划 (RC210360)
收稿时间: 2023-07-17; 修改时间: 2023-08-21; 采用时间: 2023-09-15; csa 在线出版时间: 2023-11-28
CNKI 网络首发时间: 2023-11-30

的污染. 例如水面上漂浮的塑料垃圾, 富营养藻类污染等. 污染物覆盖在水面上, 降低水中的溶解氧, 容易导致水生生物的窒息和死亡. 另外, 漂浮的污染物体影响了水面景观, 降低了水体美感, 影响旅游观赏. 及时发现并处理水面污染成为一项任务. 以往通过人工和遥感观测水面污染并清除. 人工定期巡查水面发现并清除水面污染的方法费时费力且效率低下, 而遥感对于水面漂浮的小体积污染物观测效果差且成本较高. 深度学习兴起后, 利用深度学习的方法通过岸边的摄像头或者无人船观测并清除水面污染成为更好的选择.

为解决识别水面污染物的问题, 本文利用深度学习技术提出了一种基于深度可分离卷积和交叉注意力的算法模块. 模块可以作为目标检测模型的注意力编码器起到特征增强的效果, 相比较于原始的注意力编码器, 本模块的显存占用与计算量更少. 本文研究的水面污染物是指水面上漂浮的塑料瓶等人工污染物, 使用的数据集是欧卡智舶提出的无人船视角水面漂浮物检测数据集 FloW[1]. 此数据集上的实验表明, 本模块替代模型编码器模块后在准确率与召回率上有了明显提升且显存占用大幅下降, 可以有效改善水面污染识别的算法模型.

2 相关工作

水面污染物的聚集, 对水域生态环境, 水生态系统, 渔业和旅游业等经济活动造成了负面影响[2]. 传统的图像处理和机器学习方法检测水面污染物已经得到了广泛应用, 具备较为成熟的算法模型. 胡蓉[3]使用手工设计的图像特征可以对水面污染物与背景进行有效区分. 而 Qiao 等[4]的研究表明, 传统的检测方法在处理复杂多变的场景时, 对光照, 遮挡, 形状变换等因素鲁棒性较差. 另外传统方法提取特征时较为依赖专业知识和主观经验, 可移植性差. 随着深度学习兴起, 越来越多的研究采用深度卷积神经网络识别水面污染物.

深度学习中对水面污染物的检测基于一阶段算法和二阶段算法. 一阶段算法将检测任务当成对整幅图像的回归任务, 代表算法有 YOLO[5]等. 王一早等[6]将改进 YOLO 应用于水面污染物检测的研究表明, 一阶段算法简单直接且实时性较高, 但是定位精度相对较低, 而且需要对预测的大量候选框进行后处理. 二阶段算法首先生成可能包含目标的候选框, 然后再对候选框进行精细分类和定位, 代表算法有 Mask R-CNN[7]等. 刘伟等[8]在水面污染物检测的应用表明, 二阶段算法具有较高的准确性, 对小目标和复杂场景有优势, 但是计算复杂度较高. Cheng 等[1]认为复杂的水面环境, 例如水面上的反射, 波浪等因素会干扰水面污染物识别, 且小目标污染物缺乏足够的外观信息, 难以提取深度特征. 此外, 水面污染物的识别需要借助河道摄像头[9], 遥感[10], 无人载具[11]等视角的图像数据集进行支持.

Vaswani 等[12]提出了基于自注意力机制的 Transformer 模型, 其优点是拥有强大的长距离依赖建模能力与并行计算能力. Carion 等[13]提出了 DETR 模型并实现了基于 Transformer 的端到端目标检测. DETR 取消了非极大抑制后处理且没有锚点生成. Zhang 等[14]提出了 DINO 模型, 结合了可变形注意力[15]机制和去噪训练[16]解决了 DETR 系列模型参数量大和收敛慢的问题. 相比卷积神经网络, DINO 等基于 Transformer 的网络具有强大的建模能力, 能直接从输入图像中预测目标的位置和类别且不受候选框限制, 简化了检测过程, 减少了设计复杂度. 故本文使用添加 DCCA 模块的 DINO 模型并应用到水面污染识别领域.

Sifre 等[17]提出了深度可分离卷积的卷积结构, 并被流行的模型架构 MobileNet[18]所采用. 深度可分离卷积仅沿一个空间维度应用卷积, 参数量和计算量少于标准卷积, 故本文使用深度可分离卷积减少参数量和计算量.

交叉注意力[12]作为 Transformer 的扩展技术之一, 允许模型同时对两个独立的序列进行关联性建模, 从而更好地捕捉到两个序列之间的相关信息. 本文使用交叉注意力关联不同尺度的特征图信息, 更好地识别水面污染物.

3 算法设计

本节是对深度可分离卷积和交叉注意力算法模块的设计. 本模块由跨尺度注意力融合模块与改进前馈神经网络模块组成. 本节首先介绍深度可分离卷积对比标准卷积的优势, 然后介绍跨尺度注意力融合模块与改进前馈神经网络模块, 最后介绍模块在网络上的部署.

3.1 深度可分离卷积的参数量和计算量

图 1 是标准卷积的示意图, 输入特征图大小为 $(W_{\text{in}}, H_{\text{in}})$, 输出特征图大小为 $(W_{\text{out}}, H_{\text{out}})$, 卷积核大小为 (K_w, K_h), 输入通道数为 C_{in}, 输出通道数为 C_{out}. 不考虑偏置, 参数量为 $C_{\text{in}} \times K_w \times K_h \times C_{\text{out}}$, 计算量为 $(C_{\text{in}} \times 2 \times K_w \times K_h - 1) \times W_{\text{out}} \times H_{\text{out}} \times C_{\text{out}}$.

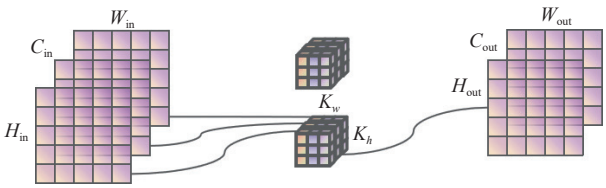

图 1 标准卷积

图 2 是深度可分离卷积示意图, 分为逐通道卷积和逐点卷积. 逐通道卷积中, 单个卷积核作用于输入的单个通道上, 这一步只改变特征图的大小. 不考虑偏置, 参数量为$C_{in}\times K_w\times K_h$. 逐点卷积中通过 1×1 卷积, 这一步改变特征图通道数. 参数量为$C_{in}\times C_{out}$. 总的参数量为$C_{in}\times K_w\times K_h+C_{in}\times C_{out}$. 计算量为$(C_{in}\times 2\times K_w\times K_h-1)\times W_{out}\times H_{out}+C_{in}\times 2\times W_{out}\times H_{out}\times C_{out}$. 可以看出深度可分离卷积参数量和计算量是标准卷积的$1/C_{out}+1/(K_w\times K_h)$.

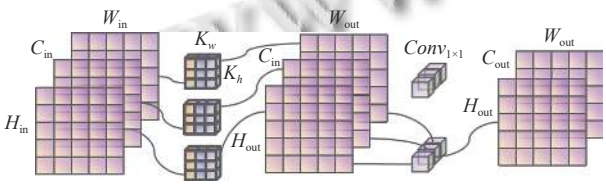

图 2 深度可分离卷积

3.2 跨尺度注意力融合模块

在神经网络中, 深层网络特征图尺寸小, 感受野大, 语义表征能力强, 但是几何信息缺乏. 而浅层网络特征图尺寸大, 感受野小, 语义表征能力弱, 含有的几何信息较多. 所以仅使用单层的特征图无法同时满足分类和定位的需求. DETR 模型仅使用最后一层特征图进行目标检测, 导致小目标物体在深层网络中的信息丢失, 所以模型检测小目标的能力差. 为了融合多个尺度的特征图, 合并深层和浅层的特征同时满足分类和定位的要求, 商汤团队提出可变形 DETR[15]. 可变形 DETR 使用了多个尺度的特征图, 结合了可变形卷积的稀疏采样能力和 Transformer 的全局建模能力聚合多尺度特征. 但是由于可变形 DETR 的编码器是在多尺度的特征图上计算自注意力并增强特征, 导致计算耗时长且参数量大. 除此之外, Lin 等[19]提出的特征金字塔网络 FPN 也是合并不同尺度特征图的方式. FPN 通过自顶向下, 横向连接的方式将不同尺度的特征图高效整合起来, 在提升检测精度的同时也没有大幅增加检测时间. 不同于 FPN 网络, 本模块使用了自底向上, 横向连接的方式整合不同尺度的特征图并增强特征, 替代 DETR 系列模型的编码器.

主干网络输出不同尺度的特征图, 随着网络深度增加, 输出的特征图尺寸缩小一半, 而通道数量增加一倍. 这样做的目的是保证网络层的复杂度. 为了拉平特征图还需对其进行通道变换, 将特征图通过 1×1 的卷积层和组归一化[20]层, 得到通道数固定的不同尺度的特征图$[X_0, X_1, \cdots, X_i, X_{i+1}]$, 将其作为跨尺度注意力融合模块的输入.

如图 3 所示, 跨尺度注意力融合模块输入是X_{i+1}和X_i^*两个不同尺度的特征图. 其中X_{i+1}是第 i+1 层的特征图, X_i^*是第 i 层特征图经过跨尺度注意力融合的输出结果. 经过交叉注意力融合后, 跨尺度注意力融合模块的输出为X_{i+1}^*.

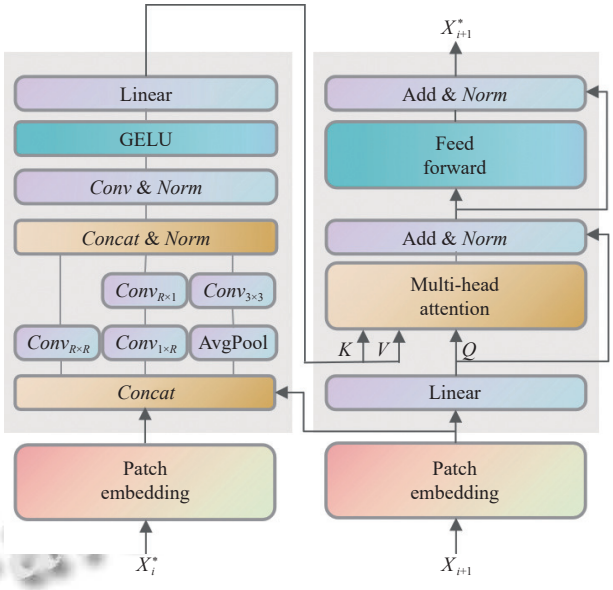

图 3 跨尺度注意力融合模块

模块内部, 将输入的特征图X_{i+1}和X_i^*进行块嵌入: 先将特征图通过卷积层, 拉平后再通过层归一化[21], 并把输出的通道数减半. 其中X_i^*需通过步长为 2 的卷积层将特征图尺寸减半. X_{i+1}通过一个全连接层后得到多头注意力层的输入查询 Q, 并把 Q 作为残差连接连到多头注意力层的输出.

Fu 等[22]提出了一种类 InceptionNet 结构的网络 IncepFormer, 通过并行的多个卷积分支获取丰富的全局上下文和精细定位特征. 与其结构类似, 本模块的输入X_{i+1}和X_i^*在块嵌入后沿通道方向进行合并, 然后将其输入到 3 个卷积分支, 如图 3 所示. 由于多头注意力层的计算复杂度与图像大小成二次方关系, 降低计算

复杂度, 需降低输入图像的分辨率. PVT[23]网络通过卷积的方式降低分辨率, 而 PVTv2[24]通过平均池化和卷积的方式降低分辨率. 用 R 表示分辨率的缩减比率, 本模块第 1 个卷积分支经过卷积核大小为 $R\times R$ 的深度可分离卷积. 第 2 个卷积分支先经过卷积核大小为 $1\times R$ 的深度可分离卷积, 再经过卷积核大小 $R\times 1$ 的深度可分离卷积, 这种带状卷积核可以更多地考虑小对象的特性. 第 3 个卷积分支先经过缩减比率为 R 的平均池, 再经过 3×3 的深度可分离卷积. 最后将 3 个分支沿通道方向合并与组归一化, 再经过激活函数后得到多头注意力层的输入键 K 和值 V. 这种结构可以融合丰富的上下文信息且不具有较大的计算复杂度. 以上过程可以表示为:

$$C_1 = DWConv_{R\times R}(X) \tag{1}$$

$$C_2 = DWConv_{R\times 1}(DWConv_{1\times R}(X)) \tag{2}$$

$$C_3 = DWConv_{3\times 3}(AvgPool_{R\times R}(X)) \tag{3}$$

$$KV = Act(Norm(Conv(Concat(C_1, C_2, C_3)))) \tag{4}$$

其中, X 是 X_{i+1} 和 X_i^* 经过块嵌入并沿通道合并的特征图, Act 为 GELU 激活函数[25].

将得到的查询 Q, 键 K 和值 V 输入多头注意力层, 计算 Q 和 K 间的相似度得到权重分布, 再与 V 做加权平均. 多头注意力考虑到序列元素间的多种相关性从而计算多次注意力并拼接得到最终输出. 多头注意力学习输入序列中各个元素间的关系从而更好地捕捉内在语义信息. 多头注意力层的输出增加了查询 Q 的残差连接[26], 目的是降低模型复杂度以减少过拟合与防止梯度消失. 归一化后输入到前馈神经网络 FFN. 前馈神经网络同样增加了输入到输出的残差连接, 归一化后得到跨尺度融合模块的输出 X_{i+1}^*. 原始的特征图序列 $[X_0, X_1, \cdots, X_i, X_{i+1}]$ 自底向上迭代多次后得到新的特征图序列 $[X_0^*, X_1^*, \cdots, X_i^*, X_{i+1}^*]$, 如图 4 所示.

3.3 改进前馈神经网络模块

多头注意力层的输出结果输入到前馈神经网络 FFN 中. 前馈神经网络首先将其通过一个全连接层进行线性变换, 再经过一个激活函数输出非线性映射, 最后经过一个全连接层输出. 由于隐藏层神经元数目大于输入层与输出层, 故前馈神经网络能将输入从一个空间映射到更高维度的空间中, 可以学习更加复杂的特征表达. 原始前馈神经网络结构如图 5 所示.

PVTv2 使用了卷积前馈神经网络, 在原始前馈神经网络的全连接层与激活函数间添加了卷积核大小为 3×3 的深度可分离卷积, 如图 6 所示, 并在深度可分离卷积中使用了 padding=1 的 0 位置填充. 这使得卷积层可以通过特征图外围填充的 0 学习到特征图的轮廓信息并减少计算量, 因此 PVTv2 取消了位置编码.

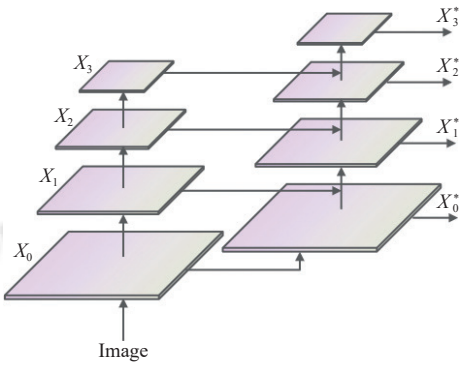

图 4 自底向上迭代特征图

图 5 原始前馈神经网络

图 6 卷积前馈神经网络

IncepFormer 中提出了一种高效前馈神经网络 E-FFN, 使用大小为 1×1 卷积代替了全连接层, 并额外添加了一个激活函数, 如图 7 所示.

图 7 高效前馈神经网络

Islam 等[27]认为 0 位置填充是 CNN 位置信息的来源, 而 padding=2 时位置信息的作用比 padding=0 和 padding=1 时更加明显. 而且更大的卷积核尺寸可能会捕获更多的位置信息. 故改进前馈神经网络模块将大小为 3×3 的卷积核换成大小为 5×5 的卷积核, 且使用 padding=2 的 0 位置填充, 如图 8 所示.

图 8 改进前馈神经网络

3.4 模块的部署

DETR 等基于 Transformer 的目标检测模型由编

码器, 解码器组成. 编码器在主干网络输出的特征图上计算自注意力, 增强图像特征. 解码器先计算目标查询间的自注意力, 以捕获目标查询间的关系, 再与编码器输出的特征图计算交叉注意力. OpenAI 推出的 GPT[28] 模型仅使用解码器的结构说明编码器可能不是必须的模块, 使用其他结构同样起到特征增强的效果. 由于编码器模块输出的特征图不改变形状, 本模块输出的特征图也不改变形状, 故用本模块直接替代编码器, 将主干网络输出的特征图序列[X_0, X_1, ⋯, X_i, X_{i+1}]输入本模块后得到新特征图序列[X_0^*, X_1^*, ⋯, X_i^*, X_{i+1}^*], 然后输入到解码器.

4 实验结果及分析

4.1 实验环境, 参数配置及评价指标

实验在百度飞桨平台使用 Python 语言完成. 平台配备的 CPU 为 Intel(R) Xeon(R) Gold 6148, 内存大小为 32 GB, 硬盘大小为 100 GB, 显卡为 Tesla v100, 显存大小为 32 GB, 操作系统版本为 Ubuntu 16.04.6 LTS, 使用的深度学习框架为 PaddlePaddle 2.4.0.

实验基于 DINO 目标检测模型, 使用的主干网络为 ResNet50, epoch 设定为 24, batch_size 设定为 2, 使用 AdamW[29]优化算法, 学习率为 0.000 1.

实验采用 COCO[30]目标检测数据集的评价指标: 平均精确率 AP, 平均召回率 AR 和 $F1$-score. 精确率与召回率的定义如式 (5) 和式 (6) 所示.

$$Precision = (TP)/(TP+FP) \tag{5}$$

$$Recall = (TP)/(TP+FN) \tag{6}$$

其中, T 与 F 表示预测是否正确, P 和 N 表示预测结果正反. 交并比 IoU 是模型输出的检测框与样本标注的真实框交集与并集的比值, 反映了目标定位算法是否精准. COCO 数据集设定了不同 IoU 阈值的判别标准. AP50:95 是 IoU 阈值在 0.5–0.95 间每隔 0.05 算 1 次平均精确率后的平均值. AP50 是 IoU 阈值大于 0.5 认为被检测到的平均精确率. AP75 是 IoU 阈值大于 0.75 认为被检测到的平均精确率. AR50:95 是 IoU 阈值在 0.5–0.95 间每隔 0.05 算 1 次平均召回率后的平均值. $F1$-score 是 AP50:95 与 AR50:95 的调和平均数, 是同时兼顾模型精确率与召回率的评价指标. FPS 是模型每秒处理的图片数量, 用来评估模型的处理速度.

4.2 数据集预处理

实验数据源自欧卡智舶发布的无人船视角水面漂浮物检测数据集 FloW. FloW 数据集由图像子数据集 FloW-Img 和多模态子数据集 FloW-RI 组成. FloW-Img 包含 2 000 张图像和 5 271 个标记目标, 其中小目标占一半以上. 该数据集使用 HDR 相机拍摄, 分辨率为 1280×720, 包含了不同场景, 光照, 水波条件的图像, 并在不同视角上对水面漂浮物进行采集.

数据集图像总的样本数为 2 000 张, 将其划分为训练集 1 600 张, 测试集 400 张. 然后将图像解码, 转换成易处理的 Numpy 格式, 然后以固定概率对图像进行翻转, 进行数据扩增. 然后对图像进行裁剪, 确保输入模型的图片数据大小统一. 再进行色调分离, 使模型能更好地根据轮廓识别图像. 然后进行归一化, 把输入图像转为固定均值, 方差的数据, 这样做的目的是使得寻找最优解的过程变得更加平滑, 使训练更容易收敛. 最后将图片数据的通道转为[C, W, H]的格式.

4.3 消融实验

为证明深度可分离卷积与交叉注意力模块的有效性, 在单类别最多有 100 个检测框, DINO 模型参数固定的情况下, 对比原始 DINO 模型, 无编码器 DINO 模型, 添加本模块的 DINO 模型的 AP50:95, AP50, AP75, AR50:95, $F1$-score, FPS 评估结果如表 1 所示.

表 1 评估结果对照

评估	原DINO	无编码器DINO	DCCA+DINO
AP50:95	0.474	0.458	**0.475**
AP50	0.887	0.869	**0.887**
AP75	0.455	0.427	**0.453**
AR50:95	0.589	0.595	**0.622**
$F1$-score	0.525	0.518	**0.539**
FPS	5.592	9.612	**8.301**

为了证明深度可分离卷积与交叉注意力模块显存占用的优势, 在输入图像尺寸为 666×1332, 其他参数固定的情况下, 对比了原始 DINO 模型, 无编码器 DINO 模型, 添加本模块的 DINO 模型的显存占用, 如表 2 所示.

可以看到添加了 DCCA 模块后 AP50 比不添加提高了 1.8%, AR50:95 提高了 2.7%. 对比原 DINO 模型, $F1$-score 提高了 0.014, FPS 提高了 2.7, 显存占用降低了 41.3%. 收敛过程中原始 DINO 模型, 无编码器 DINO 模型, 添加 DCCA 模块的 DINO 模型的 AP50 平均精确度迭代曲线与损失曲线如图 9 和图 10 所示, 可以看到算法在 24 次迭代后趋近收敛.

表 2 显存占用对照 (MB)

分类	原DINO	无编码器DINO	DCCA+DINO
输入预处理后	96.14	96.14	**96.14**
正/反向传播一次	1 673.97	251.92	**923.01**
参数	28.29	25.41	**35.71**
合计	1 798.40	373.46	**1 054.87**

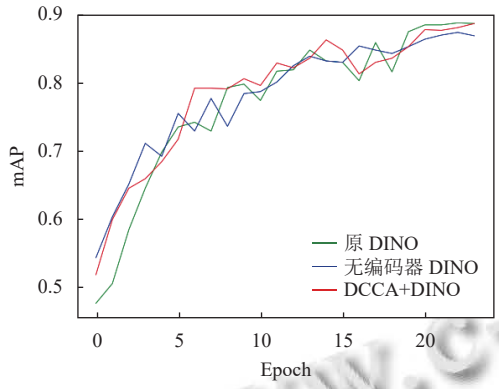

图 9 平均精确度迭代曲线

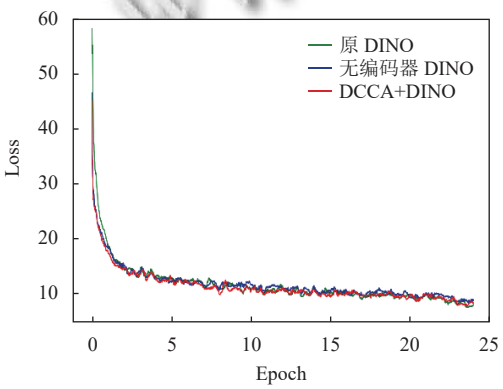

图 10 损失曲线

5 结论与展望

本文提出了基于深度可分离卷积与交叉注意力模块并将其应用到 DINO 模型, 在保证准确率与召回率的基础上解决了原有模型显存占用和计算量大的问题. 通过在 FloW 数据集上进行的水面污染识别实验表明本模型相对原模型有明显提升, 具有良好的应用空间.

参考文献

1 Cheng YW, Zhu JN, Jiang MX, *et al*. FloW: A dataset and benchmark for floating waste detection in inland waters. Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal: IEEE, 2021. 10933–10942.

2 张馨月, 高千红, 闫金波, 等. 三峡水库近坝段水面漂浮物对水质的影响. 湖泊科学, 2020, 32(3): 609–618.

3 胡蓉. 基于机器视觉的水面漂浮物自动监测的研究[硕士学位论文]. 柳州: 广西科技大学, 2015.

4 Qiao X, Bao JH, Zhang H, *et al*. fvUnderwater sea cucumber identification based on principal component analysis and support vector machine. Measurement, 2019, 133: 444–455. [doi: 10.1016/j.measurement.2018.10.039]

5 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.

6 王一早, 马纪颖, 罗星, 等. 基于 SPMYOLOv3 的水面垃圾目标检测. 计算机系统应用, 2023, 32(3): 163–170. [doi: 10.15888/j.cnki.csa.009001]

7 He KM, Gkioxari G, Dollar P, *et al*. Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386–397. [doi: 10.1109/TPAMI.2018.2844175]

8 刘伟, 王源楠, 江山, 等. 基于 Mask R-CNN 的水面漂浮物识别方法研究. 人民长江, 2021, 52(11): 226–233.

9 宋一格, 王宁, 李宏昌, 等. 基于分组卷积与双注意力机制的河流水面污染图像分类. 计算机系统应用, 2022, 31(9): 250–256. [doi: 10.15888/j.cnki.csa.008688]

10 李冠男, 王林, 李颖, 等. 卫星遥感技术在海洋倾废中的应用进展. 遥感技术与应用, 2015, 30(3): 399–406.

11 庄宝庆, 谢锡刚, 朱海, 等. 无人机遥感监测技术在河道巡查污染监测中的应用研究. 新型工业化, 2021, 11(8): 252–253.

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 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.

14 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.

15 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. Vienna: OpenReview.net, 2020.

16 Li F, Zhang H, Liu SL, *et al*. DN-DETR: Accelerate DETR training by introducing query DeNoising. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern

Recognition (CVPR). New Orleans: IEEE, 2022. 13609–13617.
17 Sifre L, Mallat S. Rigid-motion scattering for texture classification. arXiv:1403.1687, 2014.
18 Howard AG, Zhu ML, Chen B, *et al*. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017.
19 Lin TY, Dollár P, Girshick R, *et al*. Feature pyramid networks for object detection. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 936–944.
20 Wu YX, He KM. Group normalization. International Journal of Computer Vision, 2020, 128(3): 742–755. [doi: 10.1007/s11263-019-01198-w]
21 Ba JL, Kiros JR, Hinton GE. Layer normalization. arXiv: 1607.06450, 2016.
22 Fu LH, Tian HY, Zhai XP, *et al*. IncepFormer: Efficient inception transformer with pyramid pooling for semantic segmentation. arXiv:2212.03035, 2022.
23 Wang WH, Xie EZ, Li X, *et al*. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal: IEEE, 2021. 548–558.
24 Wang WH, Xie EZ, Li X, *et al*. PVTv2: Improved baselines with pyramid vision transformer. Computational Visual Media, 2022, 8(3): 415–424. [doi: 10.1007/s41095-022-0274-8]
25 Hendrycks D, Gimpel K. Gaussian error linear units (GELUs). arXiv:1606.08415, 2016.
26 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 (CVPR). Las Vegas: IEEE, 2016. 770–778.
27 Islam A, Jia S, Bruce NDB. How much position information do convolutional neural networks encode. Proceedings of the 8th International Conference on Learning Representations. Addis Ababa: OpenReview.net, 2020.
28 Radford A, Narasimhan K, Salimans T, *et al*. Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. (2018-06-11).
29 Loshchilov I, Hutter F. Decoupled weight decay regularization. Proceedings of the 7th International Conference on Learning Representations. New Orleans: OpenReview.net, 2019.
30 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.

(校对责编: 牛欣悦)