Software Engineering and Applications 软件工程与应用, 2025, 14(3), 736-750
Published Online June 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.143065

Hans汉斯

一种用于无人机小目标检测的轻量级多维特征网络

马宏伟

南京邮电大学物联网学院，江苏 南京

收稿日期：2025年4月18日；录用日期：2025年6月20日；发布日期：2025年6月30日

摘 要

无人机小目标检测在军事、搜救和智慧城市等领域有重要应用，但现有模型参数量大、计算复杂。本研究提出轻量级TriD-UAV检测器，通过构建轻量级特征提取网络TriD-Net和高效特征融合网络DENet，旨在平衡精度与计算效率。TriD-Net利用神经架构搜索引入双分支通用倒残差瓶颈结构提升效率，DENet通过深度通道部分卷积阶段减少计算并融合特征。模型使用解耦检测头和基于Wasserstein距离的损失函数增强小目标检测能力。实验表明，TriD-UAV在VisDrone数据集上实现了良好的性能与轻量化平衡，mAP50~95达到21.3%，参数量和FLOPs显著降低。

关键词

小目标检测，深度学习，轻量级网络，多维特征网络，特征融合

A Lightweight Multidimensional Feature Network for UAV Small Object Detection

Hongwei Ma

School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

Received: Apr. 18th, 2025; accepted: Jun. 20th, 2025; published: Jun. 30th, 2025

Abstract

Drone-based small object detection has significant applications in military operations, search and rescue missions, and smart city initiatives. However, existing models suffer from large parameter sizes and high computational complexity. This study proposes a lightweight TriD-UAV detector,

文章引用: 马宏伟. 一种用于无人机小目标检测的轻量级多维特征网络[J]. 软件工程与应用, 2025, 14(3): 736-750.
DOI: 10.12677/sea.2025.143065

which aims to balance accuracy and computational efficiency by constructing a lightweight feature extraction network (TriD-Net) and an efficient feature fusion network (DENet). TriD-Net employs neural architecture search (NAS) to introduce a dual-branch generalized inverted residual bottleneck structure, enhancing efficiency. DENet reduces computational overhead and fuses features through a depth wise channel-wise partial convolution stage. The model further improves small object detection capability using a decoupled detection head and a loss function based on Wasserstein distance. Experiments demonstrate that TriD-UAV achieves a favorable trade-off between performance and lightweight design on the VisDrone dataset, attaining an mAP50~95 of 21.3% while significantly reducing both parameters and FLOPs.

Keywords

Small Object Detection, Deep Learning, Lightweight Network, Multidimensional Feature Network, Feature Fusion

1. 引言

目标检测是计算机视觉中的核心任务之一，广泛应用于人脸识别、工业异常检测、吸烟检测和无人机场景应用等领域[1] [2]。基于深度学习的目标检测算法分为两种：两阶段方法和单阶段方法。两阶段算法(如 Faster R-CNN [3]和 Mask R-CNN [4])通过生成候选区域并提取特征进行分类和回归，通常精度较高，但计算开销大；单阶段算法(如 SSD [5]和 YOLO 系列[6])直接预测目标位置和类别，速度更快，但在处理小目标时，检测精度相对较低。

然而现有的无人机小目标检测模型在结构设计上仍面临诸多挑战。它们通常采用深层主干网络架构，通过堆叠几十甚至上百个卷积层来提取更抽象、更高级的特征，这不可避免地导致模型参数数量庞大。“颈部网络”处理来自主干网络不同层级的特征图，这些特征图具有显著差异：浅层特征分辨率较高，包含丰富的局部细节，而深层特征分辨率较低，含有更抽象的语义信息。为有效整合这些多样化的特征，颈部网络需要复杂的结构设计。Sayed 等人[7]通过增强小型无人机信号的信噪比，提出了 RDIwS 方法，显著提升了其检测概率和分类准确率。Hao 等人[8]提出了一种名为 SliNet 的新型框架，通过切片辅助学习神经网络将 SPPFCSPC 和 CARAFE 模块相结合，有效提升了高分辨率航空图像中小型和中型目标的检测速度与准确率。这些步骤虽然有助于特征融合，但也不可避免地增加了额外的参数和计算量，导致网络复杂性增加，从而限制了在资源受限设备上的应用，影响了无人机系统的自然应用。

本研究提出了一种轻量级的多维特征网络(DENet DPCS DBN-Loss TriD-UAV)用于无人机小目标检测。该网络包括 TriD-Net 用于特征提取，DENet 用于特征融合。TriD-Net 通过双分支跨阶段通用倒残差瓶颈结构 DeepUIB 灵活调整网络配置，以适应不同任务需求。DENet 通过部分卷积(Partial Convolution, PConv)减少冗余计算和内存访问，提高资源利用率，并增强了对图像细节和上下文信息的感知能力。最终，采用解耦检测头与 DBN-Loss 进行检测和优化，结合 Wasserstein 距离提升模型对小目标的敏感性。在小型模型比较中，TriD-UAV 表现出色。与 YOLOv10 相比，TriD-UAV(s)的 mAP50 和 mAP50~95 分别下降了 6.4%和 8.5%，但参数量和 FLOPs 分别减少了 22.2%和 29.0%。这证明了 TriD-UAV 更适合部署在资源受限的设备上，特别是在小目标检测任务中，同时保持了竞争力的准确性。

2. 相关工作

本节介绍了轻量级无人机小目标检测任务中所采用的相关技术，包括用于目标检测的轻量级网络以及边界框回归损失函数，这些技术为 TriD-UAV 的提出奠定了基础。

2.1. 目标检测轻量级网络

轻量级网络旨在计算资源受限的设备上实现高效目标检测，通过创新的网络结构和优化策略在减少模型规模和计算复杂度的同时，保持较好的检测性能。ShuffleNet [9] [10]系列包括 ShuffleNet v1 [9]和 v2 [10]，为移动设备设计。ShuffleNetv1 引入逐点分组卷积和通道洗牌机制，克服了分组卷积中信息流受限的问题，降低计算复杂度。ShuffleNetv2 在 v1 基础上优化，提供更好性能。GhostNet [11]通过 Ghost 模块减少计算量和参数数量，保持模型表达能力。MobileNet 系列[12]-[15]为移动和嵌入式视觉应用设计，MobileNetV1 [12]引入深度可分离卷积，显著降低计算复杂度；MobileNetV2 [13]增加倒残差结构和线性瓶颈，增强特征表达能力；MobileNetV3 [14]结合人工设计和神经架构搜索，优化性能与效率；MobileNetV4 [15]引入通用倒残差瓶颈模块，适应多种优化目标。FasterNet [16]提出的部分卷积(PConv)最小化冗余计算和内存访问，减少 FLOPs 并提高 FLOPS，提升处理速度和效率。

TriD-Net 通过引入双分支跨阶段通用倒残差瓶颈结构(DeepUIB)，借鉴了 MobileNet 系列中倒残差结构的思想，但通过双分支设计和神经架构搜索，使其能够更灵活地适应不同层的需求，从而在保持性能的同时提高计算效率，这与 ShuffleNet 和 MobileNet 等通过特定结构提升效率的思路一脉相承，但针对无人机小目标检测进行了定制优化。DENet 则通过深度通道部分卷积阶段(DPCS)进行高效特征融合，该模块借鉴了 FasterNet 中部分卷积的思想来减少冗余计算，但将其应用于特征融合阶段，旨在更有效地融合多尺度特征，提升复杂场景中小目标的检测精度，这与传统特征金字塔网络(如 FPN)相比，在保证融合效果的同时降低了计算开销。虽然现有网络在轻量化方面表现良好，但 TriD-Net 和 DENet 在保持相似计算复杂度的前提下，通过上述创新的特征提取与融合机制，显著提升了针对无人机小目标的检测性能。这表明本文方法在轻量化和性能之间找到了更优的平衡点，这对于资源受限的无人机平台至关重要。

2.2. 边界框回归损失函数

边界框回归损失函数在优化目标检测任务中起着关键作用。早期广泛使用的 Smooth L1 损失函数[17]结合了 L1 和 L2 损失的优点，应用于如 Faster R-CNN [3]等算法中。随着研究深入，基于 IoU 的损失函数逐渐成为主流，因为它们直接考虑预测边界框与真实边界框的重叠度，更符合目标检测的需求。尽管 IoU 损失[18]考虑了整体结构，但当预测边界框与真实边界框没有重叠时，会出现梯度消失问题。为解决此问题，广义交并比(Generalized Intersection over Union, GIoU)损失[19]引入了最小外接矩形的概念，即使边界框不重叠也能提供有效梯度，从而增强模型对不同尺度和长宽比目标的处理能力。然而，GIoU 在某些情况下仍存在收敛速度慢的问题。因此，距离交并比(Distance Intersection over Union, DIoU) [20]在 GIoU 基础上进一步引入了预测边界框和真实边界框中心点的归一化距离，提供了更直接的收敛路径，特别提升了对细长目标的检测能力。为了更全面优化边界框回归，完整交并比(Complete Intersection over Union, CIoU)损失[20]在 DIoU 基础上引入了用于调整长宽比的惩罚项，并考虑了重叠区域和中心距离。

在这些经典边界框回归损失函数的基础上，出现了许多新的回归损失函数。SCYLLA-IoU [21]引入边界框之间的向量夹角，重新定义了相关性损失函数。归一化 Wasserstein 距离(NWD) [22]是一种基于 Wasserstein 距离的小目标检测评价方法，通过计算对应高斯分布之间的相似性来衡量边界框的差异。

2.3. 无人机小目标检测算法

无人机小目标检测广泛应用于国防、军事和应急救援等领域。近年来，许多算法致力于提升其准确

性和鲁棒性。MHA-YOLOv5 [23]针对无人机图像分辨率低、信息量有限的问题，设计了多尺度混合注意力、前景增强模块和深度可分通道注意力结构，以增强小目标的特征表示能力。TriD-UAV 则在保证特征提取和融合效果的同时，更强调模型的轻量化设计，通过特定的网络结构(如 DeepUIB 和 DPC)来减少参数量和计算量，相较于 MHA-YOLO 算法，TriD-UAV 在保证效果的同时显著减少了计算开销。MFFSODNet [24]引入了额外的小目标预测头，提升了小目标检测精度并减少了参数数量。该方法还设计了多尺度特征提取模块，利用多分支卷积操作提取丰富的多尺度特征信息，并提出了双向密集特征金字塔网络，融合浅层和深层特征图的信息。TriD-UAV 采用了改进的边界框回归损失函数，使用 Wasserstein 距离增强对小目标的敏感性。E-FPN [25]针对无人机航拍图像中目标密集且混乱的特点，在主干网络中集成了简化版空间金字塔池化模块(SPP-Fast, SPPF)，提取四个尺度的特征，通过多阶段模块增强网络的目标细节捕捉能力，并通过上行和下行路径高效融合不同尺度和层级的特征信息。尽管这些算法在提高无人机小目标检测准确度方面取得了显著进展，它们通常伴随着大量的参数和计算开销。相比之下，TriD-UAV 构建了轻量级多维特征网络 TriD-Net 用于特征提取和高效表达能力网络 DENet 用于高效特征融合。TriD-Net 引入了双分支跨阶段通用倒残差瓶颈结构，提高了计算效率。DENet 则通过深度通道部分卷积阶段 DPCS 减少冗余计算和内存访问，更有效地融合空间特征。TriD-UAV 还采用了改进的边界框回归损失函数，使用 Wasserstein 距离增强对小目标的敏感性。

3. 模型设计

3.1. 轻量化多维特征网络

本章详细描述了所提出的 TriD-UAV 模型。该方法旨在尽可能保证检测精度及其他评价指标的前提下，实现模型的轻量化设计。图 1 展示了 TriD-UAV 的整体架构。

Figure 1. The overall structure of the TriD-UAV
图 1. TriD-UAV 的整体结构

在 TriD-UAV 中采用 TriD-Net 作为主干网络。如图 2 所示，TriD-Net 由五个阶段组成。初始阶段由两个卷积层组成，第二阶段由一个 DeepUIB 和一个卷积层构成。第三阶段和第四阶段具有相同的结构，每个阶段都包含一个 DeepUIB 和一个空间–通道解耦降采样(Spatial-Channel Decoupled downsampling, SCDown)。第五阶段由一个 DeepUIB、一个快速空间金字塔池化(Spatial Pyramid Pooling-Fast, SPPF)以及一个部分自注意力(Partial Self-Attention, PSA)组成。SCDown、SPPF 与 PSA 的结构如图 3~5 所示。

为了降低传统卷积中空间降采样和通道变化带来的高计算开销，SCDown 模块将这两项操作解耦：先通过 1 × 1 卷积调整通道数，再用 3 × 3 深度可分离卷积进行空间下采样，显著减少了计算量。SPPF 模块则旨在高效获取多尺度特征，通过重复应用固定大小的池化核，结合 1 × 1 卷积和拼接操作，减少重复计算，提高效率，且可学习参数量很少。

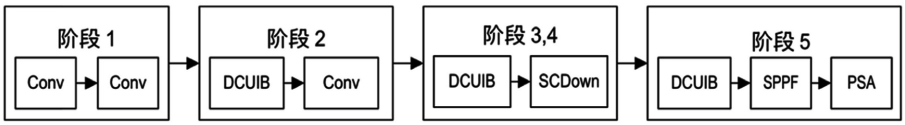

Figure 2. TriD-Net's stage composition diagram
图 2. TriD-Net 各阶段组成图

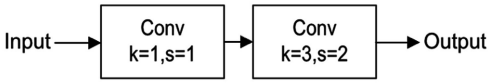

Figure 3. SCDown structure diagram
图 3. SCDown 结构图

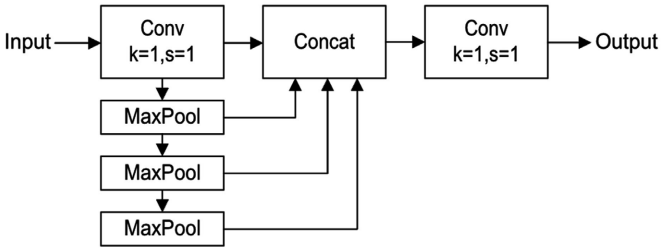

Figure 4. SPPF structure diagram
图 4. SPPF 结构图

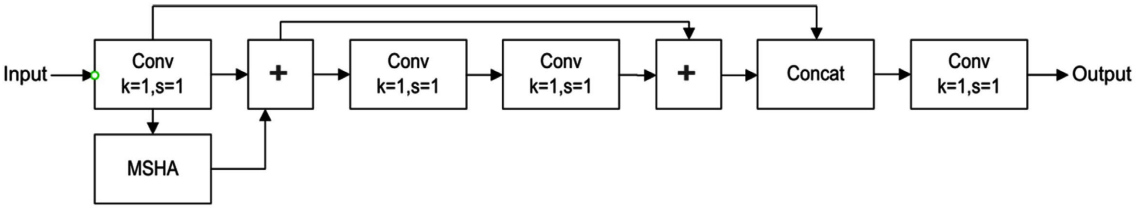

Figure 5. PSA structure diagram
图 5. PSA 结构图

为了降低自注意力机制的高计算开销，文中提出了 PSA 模块，该模块仅对分辨率最低的一半特征应用全局学习，从而在较低计算成本下实现整体特征的全局表征能力。PSA 通过将特征图分半，对其中一部分应用多头自注意力机制并与原特征融合，结合前馈网络和拼接操作，有效结合了卷积与自注意力，增强了特征表示。

输入图像经过 TriD-Net 的五个阶段进行多尺度特征提取，生成 C1 到 C5 五个特征图如图 6 所示。浅层特征图(如 C1)分辨率高，包含丰富的细节信息，但语义信息有限；深层特征图(如 C3)分辨率较低，但包含更抽象的语义信息。不同层次的特征图包含有助于目标检测的非通用信息。

Figure 6. Visualization of the five feature maps extracted by TriD-Net
图 6. TriD-Net 提取的五个特征图的可视化

TriD-Net 的核心组件是 DeepUIB，其结构如图 7 所示，对应的执行步骤见算法 1。假设输入特征的尺寸为 $H \times W \times C$，首先通过 1 × 1 卷积将通道数从 C 扩展到 $2C$，随后通过切分操作将其分成两个分支，每个分支的通道数均为 C。在分支 1 中，输入特征通过 n 个 UIB 模块堆叠，以保持特征提取能力的同时降低整体参数数量，最终输出通道数为 nC。分支 2 保持输入不变。最后，两条分支在通道维度上拼接，

从而获得尺寸为 $H\times W\times(n+1)C$ 的输出特征图。通过通道扩展，模型的特征表示能力和非线性表达能力得到了显著增强。这种扩展使得网络能够学习到更复杂、多样化的特征，捕捉更多细节和模式，从而有效避免信息瓶颈。此外，在更高维的特征空间中应用后续激活函数，进一步提升了模型对复杂数据的非线性映射能力，使其能够更准确地识别不同的形状、纹理和颜色。分支结构使得特征可以在不同路径中独立处理，从而便于并行特征学习和信息融合。每个分支均可专注于特定的特征维度，例如，一个分支可能侧重于边缘特征，而另一个分支则关注纹理特征。这种设计使模型能够充分理解输入数据，并在复杂任务中显著提高鲁棒性和灵活性。

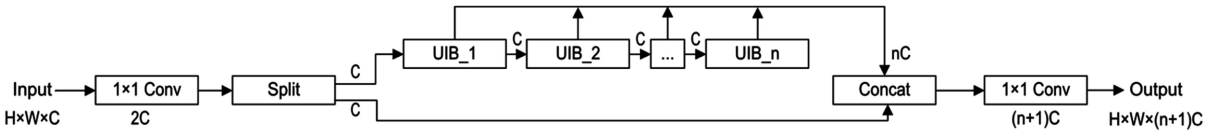

Figure 7. DeepUIB structure diagram
图 7. DeepUIB 结构图

算法 1：DeepUIB 计算过程

输入：x1(batch_size, c1, height, width)；c1；c2；n；e

输出：x2(batch_size, c2, height, width)

1：根据扩展因子 e 和目标通道数 c2 计算中间通道数 c；

2：定义一个 1x1 卷积层 cv1，将输入通道数 c1 转换为 2c，不改变特征图的尺寸。

3：定义一个 1x1 卷积层 cv2，将后续所有特征拼接后的通道数((2+n)×c)压缩至目标通道数 c2。

4：构建 n 个 UIB 模块，用于进一步提取特征。

5：将输入 x1 通过卷积 cv1 处理，得到扩展后的特征图 x。

6：将特征图 x 沿着通道维度一分为二，得到两个大小为 c 的特征块 x1 和 x2。

7：把这两个特征图放入一个列表中 y，用于后续叠加更多模块的输出。

8：逐个处理每个 UIB 模块(步骤 9-12)：

9：获取上一个最新的特征图。

11：将它输入到当前 UIB 模块中，得到新的特征图。

12：把新输出追加到 y 中。

13：将 y 中的所有特征图沿着通道维度拼接起来，得到一个通道数为((2+n)×c)的特征图。

14：通过 cv2 将拼接的特征图压缩回目标通道数 c2。

15：输出最终结果 x2。

DeepUIB 的核心组件是 UIB，其结构如图 8 所示。UIB 巧妙地扩展并改进了传统的倒残差瓶颈(Inverted Bottleneck, IB)结构。在原始 IB 模块的基础上，UIB 增加了两个可选的深度可分离卷积层：一个位于扩展层之前，另一个位于扩展层与投影层之间。神经架构搜索(Neural Architecture Search, NAS)是一种自动化方法，用于寻找最优的神经网络架构，以提升模型的性能和效率。NAS 的主要过程如图 9 所示。其主要步骤包括：定义搜索空间(层的类型、层的超参数和网络拓扑结构)，选择搜索策略，评估架构性能，优化并选择最优架构，以及进行完整的模型训练。每个生成的架构都会经过训练和评估，以确定其性能并选出最优解。经过 NAS 后，UIB 的结构依次为输入层、深度可分离卷积、线性瓶颈和输出层。

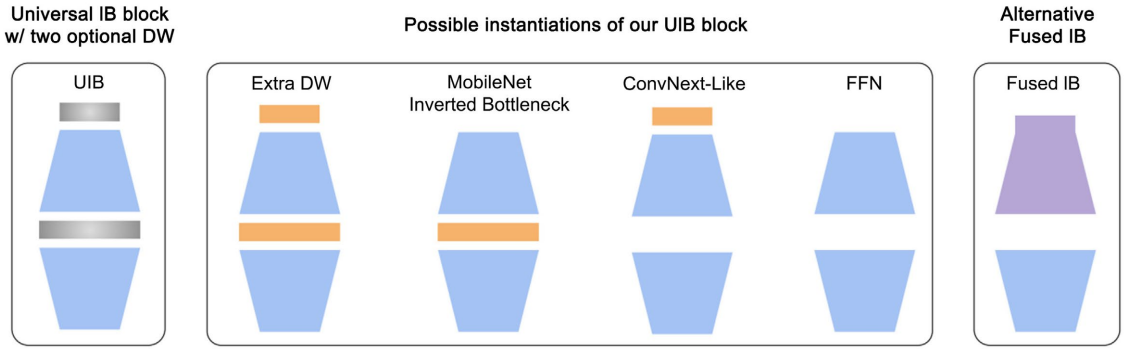

Figure 8. UIB structure diagram
图 8. UIB 结构图

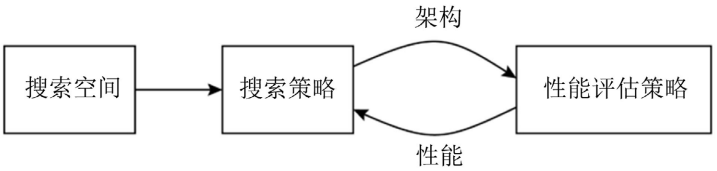

Figure 9. The main process of NAS
图 9. NAS 的主要过程

UIB 提供了在空间混合与通道混合之间灵活权衡的能力，能够根据具体需求扩展感受野，旨在最大化计算资源的利用效率。在实际应用中，UIB 的设计同样考虑到了搜索效率的问题。为避免 NAS 空间的指数级增长，UIB 共享最常见的模块(如逐点卷积的扩展和投影)，仅将深度卷积作为额外的搜索选项。该方法确保了不同实例之间可以共享大量参数，从而显著提高了 NAS 的效率。每一层的运行时间如公式(3.1)所示，其中 time 表示每一层的运行时间，$\widetilde{\text{FLOPs}}$ 表示每一层的浮点运算次数，FLOPS 表示设备的峰值浮点运算能力，ε 为设备的效率系数。

$$\text{time} = \frac{\widetilde{\text{FLOPs}}}{\text{FLOPS} \times \varepsilon} \tag{3.1}$$

3.2. 高效表达能力网络

在 TriD-UAV 中，DENet 负责融合 TriD-Net 提取的特征，生成富含语义信息的特征表示。DENet 通过处理不同尺度的特征图实现特征融合：

C5 特征：网络最深层特征，用于检测大目标，并通过上采样与低层特征融合，为其他层提供语义信息。C4 特征：连接高层语义与中层特征，融合初始 C4 特征与上采样的 C5 特征，经 DPCS 处理后用于检测中等目标，并向下传递信息。C3 特征：分辨率最高，主要用于检测小目标，融合初始 C3 特征与上采样的 C4 特征，经 DPCS 处理，并通过双向信息流保留空间细节和语义信息。

DENet 的核心是 DPCS 模块，它通过 1 × 1 卷积扩展通道，将特征分为并行路径处理，再拼接和降维，增强特征表示能力和非线性能力，同时通过分支结构促进特征多样性和融合(图 10)。

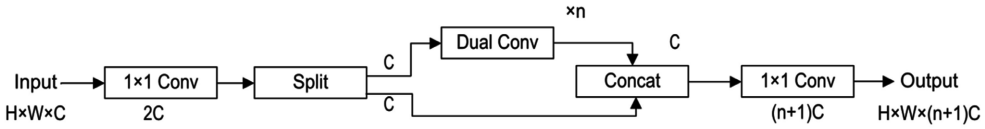

Figure 10. DPCS structure diagram
图 10. DPCS 结构图

DualConvolution 是 DPCS 中的关键结构，由两个连续的 PConv 组成，其结构如图 11 所示。PConv 的设计基于这样一种观察：在特征图的通道之间通常存在大量冗余信息。与传统对所有通道进行卷积操作的方式不同，PConv 仅对部分通道进行卷积，从而在保留关键信息的同时显著降低计算量。值得注意的是，尽管 PConv 只处理部分通道，但它仍保留了全部通道的信息。未被处理的通道可通过后续的逐点卷积继续传播信息，以确保重要特征不被丢失。这种设计使得 DPCS 能够在保持模型表达能力的同时，高效地处理和融合特征，并显著降低计算复杂度。

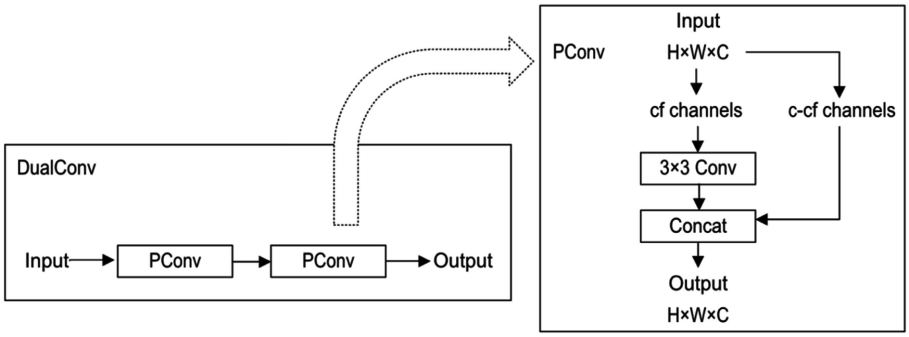

Figure 11. DualConvolution structure diagram
图 11. DualConvolution 结构图

传统卷积的 FLOPs 如公式(3.2)所示，其中 a 为所使用滤波器的卷积核大小；PConv 的 FLOPS 如公式(3.3)所示，其中 c_f 为参与计算的通道数。我们设定一个比例 r，通常为 $r=\frac{c_f}{c}=\frac{1}{4}$，则 PConv 的 FLOPs 约为传统卷积的 $\frac{1}{4}$。

$$\mathrm{FLOPs}(\mathrm{conv})=h\times w\times a^2\times c^2 \tag{3.2}$$

传统卷积和 PConv 的内存访问次数分别如公式(3.4)和(3.5)所示。当 $r=\frac{1}{4}$ 时，PConv 的内存访问次数是传统卷积的 $\frac{1}{4}$，从而显著减少了模型的参数数量和计算量。

$$\mathrm{FLOPs}(\mathrm{PConv})=h\times w\times a^2\times c_f^2 \tag{3.3}$$

$$h\times w\times 2c+a^2\times c^2\approx h\times w\times 2c \tag{3.4}$$

$$h\times w\times 2c_f+a^2\times c_f^2\approx h\times w\times 2c_f \tag{3.5}$$

3.3. 解耦头

在 TriD-UAV 中，解耦头用于检测中间特征图并输出最终结果。具体而言，对 DENet 的 DPCS_2、DPCS_3 和 DPCS_4 进行检测，从而获得最终的检测结果。与耦合检测头不同，解耦检测头将目标的位置和类别信息分开提取，分别通过不同的网络分支进行学习，最后再进行融合。

TriD-UAV 的检测头分为两部分：一对多检测头和一对一检测头。如图 12 所示，一对一检测头意味着每个检测头只负责检测一个特定类别，并输出一个边界框和对应类别。该方式适用于类别较少、目标相互独立的场景，模型结构简单，易于训练，但在复杂场景中可能难以有效捕捉类别之间的关系。一对多检测头允许同时检测多个类别，并输出多个边界框和对应的类别概率，适用于多目标检测和复杂场景，能够提升检测效率。

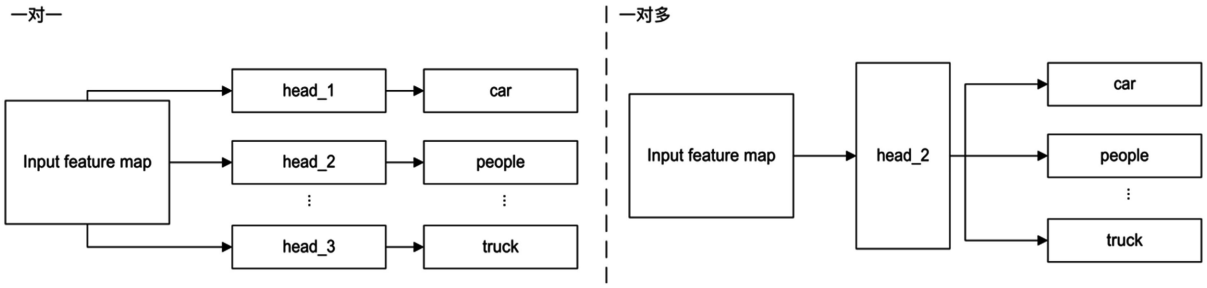

Figure 12. One-to-one detection head and one-to-many detection head
图 12. 一对一检测头和一对多检测头

在后处理阶段，首先进行置信度筛选，过滤掉所有置信度小于 0.5 的边界框；然后进行边界框坐标转换：在边框坐标表示中，(x,y) 表示中心点坐标，(w,h) 表示宽和高。使用图像坐标系时，将坐标形式 (x,y,w,h) 转换为 (x_1,y_1,x_2,y_2) 的形式，其中 (x_1,y_1) 为左上角坐标，(x_2,y_2) 为右下角坐标。最后进行类别判定，以确定目标所属的类别。

3.4. DBN 损失函数

在 TriD-UAV 中构建了 DBN-Loss 作为损失函数，包括用于分类损失的二元交叉熵损失(Binary Cross-Entropy Loss, BCE)、用于回归损失的分布焦点损失(Distribution Focal Loss, DFL [26])和归一化最优传输距离(Normalized Wasserstein Distance, NWD)。分类损失 BCE 源自信息论中的交叉熵概念，用于衡量两个概率分布之间的差异。在目标检测中，BCE 被用来衡量预测分布与真实分布之间的差异。BCE 的计算如公式(3.6)所示，其中 y_l 表示真实标签，P_l 表示模型预测的概率。

$$L_{\mathrm{BCE}}=-\left(y_l\log P_l+(1-y_l)\log(1-P_l)\right) \tag{3.6}$$

通常情况下，假设的狄拉克 δ 分布位于预测值 v_p 真实标签 l_{gt} 之间，如公式(3.7)所示。然后将其乘以 v_p 将恢复标签 l_{gt}，如公式(3.8)所示。现在，使用 $O(v_p)$ 直接表示广义分布，假设标签 $\widehat{l_{gt}}\in\left[l_{gt0},l_{gtn}\right]$，并且可以根据模型进行估计，如公式(3.9)所示。然后，连续积分被离散化表示，并且离散概率和为 1 的性质也得到了保证，如式(3.10)所示。所以公式(3.9)可以表示为公式(3.11)。将 $O(v_p)$ 简化为具有 $n+1$ 个单元的 Softmax 层，则 DFL 的计算公式如式(3.12)所示。

$$\int_{-\infty}^{+\infty}\delta(v_p-l_{gt})\mathrm{d}v_p=1 \tag{3.7}$$

$$l_{gt}=\int_{-\infty}^{+\infty}\delta(v_p-l_{gt})v_p\mathrm{d}v_p=1 \tag{3.8}$$

$$\widehat{l_{gt}}=\int_{-\infty}^{+\infty}O(v_p)v_p\mathrm{d}v_p=\int_{l_{gt0}}^{l_{gtn}}O(v_p)v_p\mathrm{d}v_p \tag{3.9}$$

$$\sum_{i=0}^{n}O(l_{gti})=1 \tag{3.10}$$

$$\widehat{l_{gt}}=\sum_{i=0}^{n}O(l_{gti})l_{gti} \tag{3.11}$$

$$\mathrm{DFL}(S_i,S_{i+1})=-\left((l_{gti+1}-l_{gt})\log(S_i)+(l_{gt}-l_{gti})\log(S_{i+1})\right) \tag{3.12}$$

对于水平边界框 $C=(x_c,y_c,w,h)$，其中 (x_c,y_c) 为中心坐标，w 为宽度，h 为高度。其内在椭圆的方程可以表示为式(3.13)。其中 (α_x,α_y) 为椭圆的中心坐标，λ_x 和 λ_y 分别为沿 x 轴和 y 轴的半轴长度，且满

足：$\alpha_x = x_c, \alpha_y = y_c, \lambda_x = \frac{w}{2}, \lambda_y = \frac{h}{2}$。二维高斯分布的概率密度函数如式(3.14)所示，其中 c,σ,τ 分别表示坐标 (x,y)、均值向量和高斯分布的协方差矩阵。

$$\frac{(x-\alpha_x)^2}{\lambda_x^2} + \frac{(y-\alpha_y)^2}{\lambda_y^2} = 1 \tag{3.13}$$

$$f(c \mid \sigma,\tau) = \frac{\exp\left(-\frac{1}{2}(c-\sigma)^{\mathrm{T}} \tau^{-1}(c-\sigma)\right)}{2\pi|\tau|^{\frac{1}{2}}} \tag{3.14}$$

当 c,σ,τ 满足式(3.15)中的关系时，式(3.13)中的椭圆将成为二维高斯分布的密度轮廓。因此，水平边界框 $C=(x_c,y_c,w,h)$ 可以被建模为二维高斯分布 $Q(\sigma,\tau)$，如式(3.16)所示。两个边界框之间的相似性可以转化为两个高斯分布之间的分布距离。对于两个二维高斯分布 $\sigma_1 = Q_1(n_1,\tau_1)$ 和 $\sigma_2 = Q_2(n_2,\tau_2)$，$\sigma_1$ 和 σ_2 之间的二阶瓦瑟斯坦距离可以表示为式(3.17)，并且可以简化为式(3.18)，其中 $\|\cdot\|_F$ 表示弗罗贝尼乌斯范数。此外，对于由边界框 $A=(x_a,y_a,w_a,h_a)$ 和 $B=(x_b,y_b,w_b,h_b)$ 建模的高斯分布 Q_a 和 Q_b，式(3.18)可以进一步简化为式(3.19)。但是，$D_2^2(Q_a,Q_b)$ 是一种距离度量，不能直接作为相似性度量。因此通过对其指数形式进行归一化，得到 NWD，如式(3.20)所示。其中 D 是与数据集密切相关的常数。将 NWD 作为损失函数，如式(3.21)所示，其中 Q_p 和 Q_{gt} 分别是为预测边界框和真实边界框建模的高斯分布。

$$(c-\sigma)^{\mathrm{T}} \tau^{-1}(c-\sigma) = 1 \tag{3.15}$$

$$\sigma = \begin{bmatrix} x_c \\ y_c \end{bmatrix}, \tau = \begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix} \tag{3.16}$$

$$D_2^2(\sigma_1,\sigma_2) = \|n_1 - n_2\|_2^2 + Tr\left(\tau_2 + \tau_1 - 2\left(\tau_2^{\frac{1}{2}}\tau_1\tau_2^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) \tag{3.17}$$

$$D_2^2(\sigma_1,\sigma_2) = \|n_1 - n_2\|_2^2 + \left\|\tau_1^{\frac{1}{2}} - \tau_2^{\frac{1}{2}}\right\|_F^2 \tag{3.18}$$

$$D_2^2(Q_a,Q_b) = \left\|\left[x_a, y_a, \frac{w_a}{2}, \frac{h_a}{2}\right]^{\mathrm{T}}, \left[x_b, y_b, \frac{w_b}{2}, \frac{h_b}{2}\right]^{\mathrm{T}}\right\|_2^2 \tag{3.19}$$

$$\mathrm{NWD}(Q_a,Q_b) = \exp\left(-\frac{\sqrt{D_2^2(Q_a,Q_b)}}{D}\right) \tag{3.20}$$

$$L_{\mathrm{NWD}} = 1 - \mathrm{NWD}(Q_p,Q_{gt}) \tag{3.21}$$

4. 实验实现与分析

本节全面总结了实验工作，利用多种目标检测评估指标(如精度和参数量)对结果进行分析，并通过表格和图形的形式展示，以验证我们所提出的方法。实验使用了 VisDrone 数据集，该数据集的主要特点是小目标占比较高，分布比例为：小目标 68.4%，中目标 28.7%，大目标 2.9%。目标大小根据边界框尺寸

划分：小目标小于 32 × 32 像素，中目标介于 32 × 32 和 96 × 96 像素之间，大目标大于 96 × 96 像素。VisDrone 数据集共包含 8599 张图像，涵盖十个类别，并划分为训练集(6471 张)、验证集(548 张)和测试集(1580 张)。数据集图像来源于不同地点但环境相似，具有多样性和一致性。

目标检测任务的检测结果可以分为四类。精度(Precision, P)和召回率(Recall, R)可以表示模型的准确性。平均精度(Average-Precision, AP)表示检测器在每个召回率下的平均精度，平均精度均值(mean Average Precision, mAP)表示所有类型的 AP 的平均值，mAP50 表示 IoU 阈值为 0.5 时的 mAP，mAP50~95 表示在多个 IoU 阈值从 0.5 到 0.95，步长为 0.05 下的平均 mAP。模型的复杂度通常通过两个指标来衡量：参数和 FLOPs。参数指的是模型中需要训练的总参数数量，对应于空间复杂度。FLOPs 用于衡量算法和模型的计算复杂度，通常作为模型速度的间接度量，对应于时间复杂度。

所有实验均在配备 RTX3090(24GB)GPU 和 14 个 vCPU 的 Intel(R) Xeon(R) Platinum 8362 CPU@2.80GHz 的环境中进行。模型从零开始训练，未使用任何预训练权重。具体的实验参数设置见表 1。这些训练参数确保了模型的充分收敛，同时避免了过拟合。与学习率相关的参数可以使模型稳定地执行。数据增强方法可以提高数据的多样性。损失函数权重可以为不同任务分配重要性。

Table 1. The parameter settings used in the experiment.
表 1. 实验中使用的参数设置

类别	具体项目	数值
训练参数	epochs	200
	batch	16, 32, 64
	image-size	640
	optimizer	SGD
学习率相关	lr0	0.01
	lrf	0.01
数据增强	mosaic	1.0
	fliplr	0.5
损失函数权重	box	7.5
	cls	0.5
	dfl	1.5

4.1. 在 VisDrone 数据集上的实验结果

这一部分主要验证提出的 TriD-UAV 的有效性，并在 VisDrone 数据集上与多个模型进行比较。表 2 显示了 TriD-UAV 系列模型与 YOLO 系列模型在 VisDrone 数据集上的比较，结果表明，TriD-UAV 在准确性和复杂性之间达到了优异的平衡。TriD-UAV(n)在轻量级模型比较中表现出了显著的效率优势。与 YOLOv8(n)相比，TriD-UAV(n)的 mAP50 和 mAP50~95 分别下降了 9.7%和 11.0%，但参数量减少了 33.3%，FLOPs 减少了 21.0%。与 YOLOv10(n)相比，mAP50 降低了 5.7%，mAP50~95 降低了 7.6%，参数量减少了 25.9%，FLOPs 减少了 23.8%。这表明，TriD-UAV(n)显著降低了模型复杂性，并在牺牲部分准确度的情况下，达到了更好的效率平衡。

在小型模型比较中，TriD-UAV(s)也表现出色。与 YOLOv10(s)相比，TriD-UAV(s)的 mAP50 和 mAP50~95 分别下降了 6.4%和 8.5%，但参数量和 FLOPs 分别减少了 22.2%和 29.0%。值得注意的是，与

Mamba-YOLO(t)相比，TriD-UAV(s)在相似的复杂度下实现了更高的mAP50和mAP50~95。这证明了TriD-UAV(s)更适合部署在资源受限的设备上，特别是在小目标检测任务中，同时保持了竞争力的准确性。

Table 2. Comparison of accuracy and complexity between the TriD-UAV series models and YOLO series models on the VisDrone dataset
表 2. TriD-UAV 系列模型与 YOLO 系列模型在 VisDrone 数据集上的精度与复杂度对比

模型	P(%)	R(%)	mAP50(%)	mAP50~95(%)	Para.(M)	FLOPs(G)
YOLOv3	59.0	51.8	53.1	32.5	61.5	155.4
YOLOv5	64.2	53.4	56.4	35.1	119.0	171.4
YOLOv7	63.7	57.5	58.1	35.0	37.3	105.3
YOLOv8(n)	43.8	32.8	32.9	19.0	3.0	8.1
YOLOv8(s)	49.7	39.4	40.2	23.9	11.1	28.5
YOLOv10(n)	41.5	32.2	31.5	18.3	2.7	8.4
YOLOv10(s)	50.2	37.8	38.9	23.3	8.1	24.8
Mamba-YOLO(t)	45.0	34.1	34.3	20.0	6.0	13.6
TriD-UAV(n)	40.8	30.5	29.7	16.9	2.0	6.4
TriD-UAV(s)	46.7	36.0	36.4	21.3	6.3	17.6

此外，TriD-UAV 系列展现了出色的可扩展性。从 TriD-UAV(n)到 TriD-UAV(s)，参数量仅增加了 4.3 M，FLOPs 增加了 11.2 G，同时准确性指标稳步提高。这种高效的规模扩展为不同场景下的无人机小目标检测提供了灵活多样的选择。

表 3 比较了 TriD-UAV 与其他方法在 VisDrone 数据集中不同类别的 mAP50~95 值。结果表明，尽管 TriD-UAV 是一个轻量级模型，但它在多个目标类别中表现令人满意。具体而言，TriD-UAV 在行人、汽车和自行车等常见类别中的表现优异，mAP50~95 值与一些更复杂的模型相当。值得注意的是，TriD-UAV 的整体 mAP50~95 值也达到了相当高的水平，证明了其在整体检测性能上的有效性。

Table 3. Comparison of mAP50~95 values for different models across categories on the VisDrone dataset
表 3. 不同模型在 VisDrone 数据集上各类别的 mAP50~95 值对比

模型	Ped	Peo	Bic	Car	Van	Truck	all
YOLOv3	22.0	13.7	7.1	53.9	30.4	25.0	23.3
YOLOv5	23.4	15.4	7.8	58.9	34.6	30.9	27.3
YOLOv7	24.4	18.1	8.4	57.5	34.2	28.1	26.9
YOLOv8(n)	23.1	15.5	7.9	58.9	35.4	30.4	27.2
YOLOv10(n)	13.7	10.2	3.4	50.7	25.2	17.3	18.3
YOLOv10(s)	18.8	12.9	5.8	56.3	31.2	23.6	23.3
Mamba-YOLO(t)	15.6	10.6	3.6	53.1	28.3	19.1	20.0
TriD-UAV(n)	12.9	9.7	2.9	49.5	23.4	13.6	16.9
TriD-UAV(s)	17.1	12.1	4.5	54.6	29.2	20.2	21.3

参数和 6.4 G 计算量，达到了 40.8%的检测精度。这个性能显著优于具有相似参数数量的轻量级模

型，如 ConvNeXt 和 EfficientNet。从模型效率的角度来看，TriD-UAV(n)的参数数量仅为 2.0 M，与表中最轻量的 VanillaNet 和 ShuffleNetV2 相当，但其检测性能明显领先。同时，尽管像 PANet 和 BiTFN 这样的模型获得了稍高的 mAP50，它们的参数数量和计算负载远高于我们的方法，在实际部署中可能面临更大的资源压力。这些对比结果充分展示了 TriD-UAV(n)在模型轻量化和性能平衡方面的优势，特别适合在计算资源有限的无人机平台上进行部署和应用。表 4 显示了 TriD-UAV 与其他轻量级 UAV 目标检测算法的比较。

Table 4. Comparison experiment of TriD-UAV(n) with other lightweight backbones and neck structures on the VisDrone dataset
表 4. TriD-UAV(n)与其他轻量级骨干网和颈部结构在 VisDrone 数据集上的对比实验

模型	Para.(M)	FLOPs(G)	P(%)	R(%)	mAP50(%)	mAP50~95(%)	Aps(%)
EfficientNet	2.1	5.9	35.4	26.3	24.8	14.0	6.2
ConvNeXt	2.0	5.3	28.8	21.7	19.9	11.1	4.6
VanillaNet	1.4	3.6	27.2	20.4	18.0	9.7	2.9
ShuffleNetV2	1.9	5.2	31.4	22.5	20.6	11.5	4.6
PANet	4.0	9.1	42.6	31.1	31.1	18.1	8.6
BiFPN	3.2	8.3	41.6	31.1	30.7	17.9	8.5
MobileNetV3	2.6	5.9	34.3	24.6	22.9	12.8	5.7
FPN	4.8	19.1	41.6	32.0	31.4	17.8	8.7
TriD-UAV(n)	2.0	6.4	40.8	30.5	29.7	16.9	7.0

4.2. 消融实验

消融实验在 VisDrone 数据集上进行。表 5 显示，与耦合头相比，解耦头在保持可接受的检测性能的同时实现了显著的轻量化。特别地，解耦头的参数数量和计算量相比耦合头减少了近 50%。尽管解耦头在检测指标上有所下降，但考虑到无人机场景对模型轻量化的迫切需求，这一性能损失是可以接受的。这些结果表明，分离检测任务的解耦设计有效提高了模型效率，更适合在计算资源有限的无人机平台上部署。

Table 5. Validation of the effectiveness of different detection heads
表 5. 不同检测头的有效性验证

检测头	P(%)	R(%)	mAP50(%)	mAP50~95(%)	Para.(M)	FLOPs(G)
Coupled Head	38.2	27.3	28.1	15.7	5.3	4.5
Decoupled Head	39.8	30.2	29.7	17.0	2.7	8.4

表 6 讨论了 DeepUIB、DPCS 和 DBN-Loss 模块对模型性能和计算效率的影响。DeepUIB 是 TriD-Net 的重要组成部分，DPCS 是 DE-Net 的重要组成部分。实验结果表明，结合这些模块显著提高了计算效率，同时保持了接近基准性能。

目前，最终的组合模型在性能指标上与基准相比仅表现出轻微的偏差。其精度略有下降，从 39.8%下降到 38.3%，下降了 1.5%。召回率也有所下降，从 30.2%降至 28.2%，下降了 2.0%。mAP50 指数出现了轻微下降，从 29.7%降至 28.5%，下降幅度为 3.37%，而 mAP50~95 指数则从 17.0%降至 15.9%，下降了 5.29%。尽管这些下降幅度较小，但组合模型始终保持了与基准相当的性能水平。

Table 6. Validation of the effectiveness of each module.
表 6. 各模块有效性验证

模块	P(%)	R(%)	mAP50(%)	mAP50~95(%)	Para.(M)	FLOPs(G)
Baseline	39.8	30.2	29.7	17.0	2.7	8.4
DeepUIB	37.0	28.2	27.0	15.3	2.3	7.2
DPCS	38.0	29.6	27.9	15.8	2.4	7.7
DBN-Loss	39.3	29.6	28.6	16.3	2.7	8.4
DeepUIB + DPCS + DBN-Loss	39.2	29.6	28.7	16.1	2.0	6.4

更值得注意的是计算资源的节省。当前的组合模型将参数数量减少了 25.9%，从 2.7 M 降至 2.0 M。类似地，FLOPs 减少了 23.8%，从 8.4 G 降至 6.4 G。在计算资源有限的环境中部署模型时，这种显著的资源节省至关重要。

通过不断分析各个模块的独立贡献，发现 DeepUIB 和 DPCS 虽然在有效减少计算资源的同时，但是却导致了性能的轻微下降。而 DBN-Loss 则巧妙地抑制了性能损失，同时没有增加额外的计算负担。这种和谐的组合充分发挥了各个模块的优势，达到了资源利用和性能之间的良好平衡。

5. 结论

在本研究中提出了一种用于小物体检测任务的轻量级无人机检测器——轻量级多维特征网络(TriD-UAV)，旨在减少无人机小物体检测任务中的参数数量和计算复杂度。TriD-UAV 的核心创新包括构建轻量级多维特征网络，通过双分支跨阶段通用倒置瓶颈模块灵活筛选网络层以适应不同功能，设计高效表达网络，使用通道级部分卷积阶段在显著保持准确性的同时减少模型复杂度，并提出 DBN-Loss 以提高目标边界框的定位精度。实验结果表明，TriD-UAV 在多个数据集上表现良好，在牺牲少量精度的情况下显著减少了模型的计算量，使其能够有效地部署在资源受限的设备上。然而，在红外图像处理方面，TriD-UAV 的准确性显著下降，这可能是因为红外图像包含更多的细节信息，当前的结构难以全面捕捉。针对这一问题，我们计划在未来的工作中引入知识蒸馏技术，使用大型模型作为教师网络，引导 TriD-UAV 学习更丰富的特征表示。

参考文献

[1] Xue, Y., Jin, G., Shen, T., Tan, L., Wang, N., Gao, J., *et al*. (2023) SmallTrack: Wavelet Pooling and Graph Enhanced Classification for UAV Small Object Tracking. *IEEE Transactions on Geoscience and Remote Sensing*, **61**, 1-15. https://doi.org/10.1109/tgrs.2023.3305728

[2] Xue, Y., Jin, G., Shen, T., Tan, L. and Wang, L. (2023) Template-Guided Frequency Attention and Adaptive Cross-Entropy Loss for UAV Visual Tracking. *Chinese Journal of Aeronautics*, **36**, 299-312. https://doi.org/10.1016/j.cja.2023.03.048

[3] Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv: 1506.01497.

[4] He, K., Gkioxari, G., Dollar, P. and Girshick, R. (2017) Mask R-CNN. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 2980-2988. https://doi.org/10.1109/iccv.2017.322

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., *et al*. (2016) SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[6] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 779-788 https://doi.org/10.1109/cvpr.2016.91

[7] Sayed, A.N., Ramahi, O.M. and Shaker, G. (2024) RDIwS: An Efficient Beamforming-Based Method for UAV Detection and Classification. *IEEE Sensors Journal*, **24**, 15230-15240. https://doi.org/10.1109/jsen.2024.3375862

[8] Hu, N., Yang, J., Pan, W., Xu, Q., Shao, S. and Tang, Y. (2024) UAV Detection Based on the Variance of Higher-Order Cumulants. *IEEE Transactions on Vehicular Technology*, **73**, 11182-11195. https://doi.org/10.1109/tvt.2024.3370590

[9] Zhang, X., Zhou, X., Lin, M. and Sun, J. (2018) ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 6848-6856. https://doi.org/10.1109/cvpr.2018.00716

[10] Ma, N., Zhang, X., Zheng, H. and Sun, J. (2018) ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In: Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, 122-138. https://doi.org/10.1007/978-3-030-01264-9_8

[11] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C. and Xu, C. (2020) GhostNet: More Features from Cheap Operations. 2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 13-19 June 2020, 1577-1586. https://doi.org/10.1109/cvpr42600.2020.00165

[12] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv: 1704.04861.

[13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/cvpr.2018.00474

[14] Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., *et al*. (2019) Searching for MobileNetV3. 2019 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Seoul, 27 October-2 November 2019, 1314-1324. https://doi.org/10.1109/iccv.2019.00140

[15] Qin, D., Leichner, C., Delakis, M., Fornoni, M., Luo, S., Yang, F., *et al*. (2024) MobileNetV4: Universal Models for the Mobile Ecosystem. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T. and Varol, G., Eds., *Computer Vision—ECCV* 2024, Springer, 78-96. https://doi.org/10.1007/978-3-031-73661-2_5

[16] Chen, J., Kao, S., He, H., Zhuo, W., Wen, S., Lee, C., *et al*. (2023) Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. 2023 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Vancouver, 17-24 June 2023, 12021-12031. https://doi.org/10.1109/cvpr52729.2023.01157

[17] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/iccv.2015.169

[18] Yu, J., Jiang, Y., Wang, Z., Cao, Z. and Huang, T. (2016) UnitBox: An Advanced Object Detection Network. *Proceedings of the* 24*th ACM International Conference on Multimedia*, Amsterdam, 15-19 October 2016, 516-520. https://doi.org/10.1145/2964284.2967274

[19] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. and Savarese, S. (2019) Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 658-666. https://doi.org/10.1109/cvpr.2019.00075

[20] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R. and Ren, D. (2020) Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

[21] Gevorgyan, Z. (2022) SIoU Loss: More Powerful Learning for Bounding Box Regression. arXiv: 2205.12740.

[22] Wang, J., Xu, C., Yang, W. and Yu, L. (2021) A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. arXiv: 2110.13389.

[23] Song, G., Du, H., Zhang, X., Bao, F. and Zhang, Y. (2024) Small Object Detection in Unmanned Aerial Vehicle Images Using Multi-Scale Hybrid Attention. *Engineering Applications of Artificial Intelligence*, **128**, Article ID: 107455. https://doi.org/10.1016/j.engappai.2023.107455

[24] Jiang, L., Yuan, B., Du, J., Chen, B., Xie, H., Tian, J., *et al*. (2024) MFFSODNet: Multiscale Feature Fusion Small Object Detection Network for UAV Aerial Images. *IEEE Transactions on Instrumentation and Measurement*, **73**, 1-14. https://doi.org/10.1109/tim.2024.3381272

[25] Li, Z., He, Q. and Yang, W. (2024) E-FPN: An Enhanced Feature Pyramid Network for UAV Scenarios Detection. *The Visual Computer*, **41**, 675-693. https://doi.org/10.1007/s00371-024-03355-w

[26] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J. and Yang, J. (2020) Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *Advances in Neural Information Processing Systems*, **33**, 21002-21012.