3 基于 Ghost 改进的 YOLOv5 检测算法

3.1 引言

YOLO 检测算法因为其检测速度快而准，所以被广泛应用于需要实时处理的场景。然而，原始的 YOLO 模型可能对于资源受限的环境（如移动设备或嵌入式系统）来说仍然太大或太复杂。因此，本节基于 YOLOv5s 检测算法进行了改进，设计了一种轻量化的网络模型，并且提高了一定检测精度。在自制数据集上表现良好。

3.2YOLOv5s 算法

YOLOV5 是 YOLO 系列第 5 代目标检测模型，由 ultralytics 团队开发，其中根据网络深度和宽度的不同，可以分为 s、m、l 和 x 版本。YOLOv5s 模型[38]的网络深度和宽度都比较小。它以其轻量级的设计、出色的性能和易用性而受到广泛应用。是本文所改进模型的基础。

YOLOv5s 网络主要由 3 部分构成，Backbone（主干网）、Neck（颈部）和 Head（检测头）构成。YOLOv5s 网络结构如图 3-1 所示。主干网络主要负责从输入图像中提取多尺度特征。由 CBS（Conv+BN+SiLU）、C3 模块和 SPPF（Spatial Pyramid Pooling - Fast）快速空间金字塔池化构成[39]。SiLU 激活函数是 ReLU 激活函数的一种变种，它将 Sigmoid 函数和 ReLU 函数结合起来。SiLU 在大于等于 0 时，提供了正的梯度，这有助于缓解梯度消失问题，尤其是在深层网络中；在小于等于时输出均值为 0，这有助于保持模型的稳定性，并减少了对输入数据分布的偏移。其中 C3 模块作为主要结构，这是 YOLOv5s 的核心特征提取部分，C3 模块中的 BottleNeck 结构减少计算复杂度，同时保持或提高模型性能[40]。SPPF 通过使用多个最大池化操作，在不同的尺度上对特征图进行池化，然后拼接特征图，从而增加网络对不同尺度特征的感知能力。这使得网络能够更有效地捕捉到图像中的目标信息和上下文关系[41]。

颈部网络负责将主干网络提取的特征进行级联和融合，以形成更丰富的特征表示。YOLOv5s 的颈部网络部分主要包括 PANet 结构，它能够有效地将不同层次的特征图融合，以提供更好的多尺度信息。PANet 结构如图 3-2 所示。FPN 是一个自顶向下的特征融合框架，它通过上采样操作将高层的特征图映射到与原始图像相同的尺

寸，同时通过横向连将高层特征与低层特征进行融合，确保了语义信息与位置信息的有效结合。在 FPN 的基础上，PANet 增加了一个自底向上的路径，这个路径通过一系列卷积操作增强低层特征，并将其高效地传递到高层，使得低层的位置信息能够更好地辅助高层的语义信息。

检测头包含预测层，负责输出每个目标的类别概率、位置坐标和置信度。输出层使用了 YOLOv5s 独特的 Detect 层，该层会处理特征图并输出检测结果。

YOLOv5s 在训练和测试时，采用了 Mosaic 数据增强策略，以此丰富训练数据并提升模型的泛化性能。此外，YOLOv5s 还引入了自适应锚框的概念，它允许模型自动学习和调整锚框的大小和比例，以更好地适应不同尺寸和比例的目标。

在输入输出方面，YOLOv5s 接受尺寸为 640x640 的图像，并输出一个张量，其中包含检测框的位置、大小、类别概率和目标置信度。这些输出可用于后续的目标识别和跟踪任务。

YOLOv5s 模型的大小和计算复杂度相对较小，这使得它非常适合在资源受限的环境中部署，如嵌入式设备或无人机。其高性能和准确性也使其成安全帽与反光衣检测任务的首选模型。

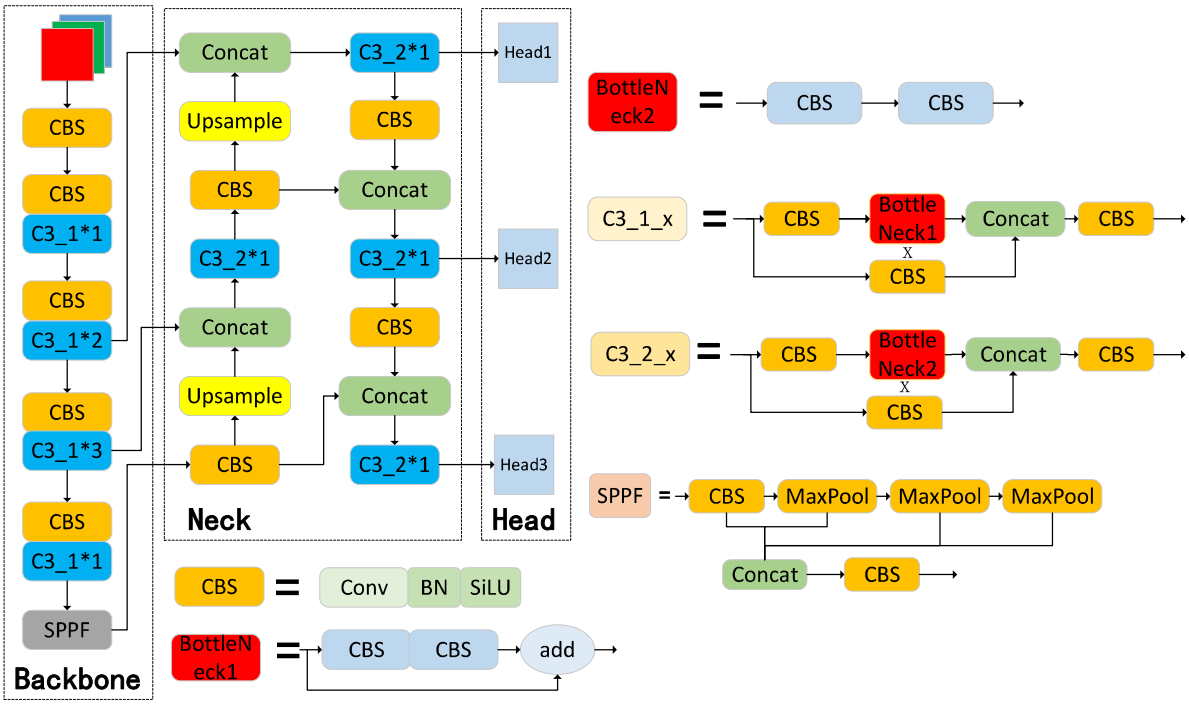


图 3-1 YOLOv5s 网络结构

Fig. 3-1 YOLOv5s network structure

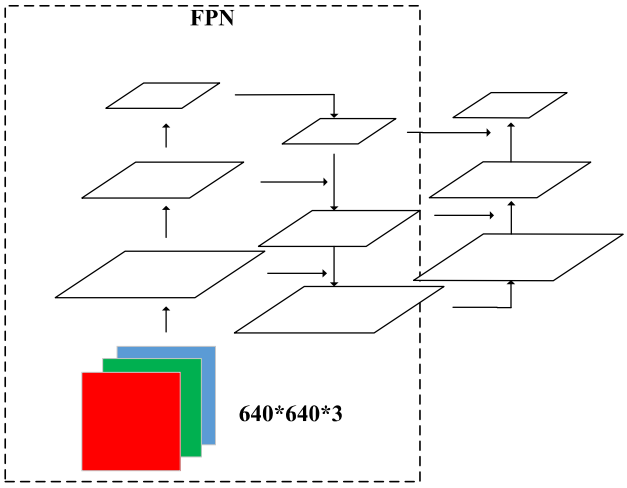


图 3-2 PANet 网络结构

Fig. 3-2 PANet network structure

3.3 改进的 YOLOv5s 算法

3.3.1 Ghost 模块的引入

在实际的应用环境中，YOLOv5s 网络模型性能容易受到硬件内存与计算量的影响。为了适应移动与嵌入型设备，引入了 GhostNet[42]网络结构，以降低模型复杂度。GhostNet 是一种轻量化的网络，与传统的卷积神经网络相比，GhostNet 能够通过较少的计算成本生成更多的特征图。GhostNet 网络主要由 Ghost 模块构成[43]。

Ghost 模块将传统的卷积与简单的线性运算相结合，利用了特征提取与特征图冗余的关系[44]。Ghost 模块结构如图 3-3。在 YOLOv5s 网络中，Ghost 模块代替了传统了卷积过程，先对特征图进行普通卷积操作，进行通道压缩，然后进行线性变换获得更多的特征图，然后将得到的特征图进行拼接操作，得到最终的特征图，其中 Φ 表示线性变换[39]。普通卷积浮点计算量 FLOPs 的公式如下：

$$F_1 = n \times h' \times w' \times c \times g \times g \tag{3.1}$$

Ghost 模块的浮点计算量 FLOPs 公式如下：

$$F_2 = \frac{n}{s} \times h' \times w' \times c \times g \times g + (s-1) \times h' \times w' \times \frac{n}{s} \times d \times d \tag{3.2}$$

其中，n 为输出通道的数量；h' 输出特征的高度；w' 为输出特征的宽度；c 为输入通道的数量；g 为卷积核大小；s 为 Ghost 模块中生成的特征图数量；d 为线性操作卷

积核的大小。

$$\frac{F_1}{F_2}=\frac{c\times g\times g}{(1/s)\times c\times g\times g+d\times d\times(s-1)/s}\approx\frac{s\times c}{s+c-1}\approx s \tag{3.3}$$

其中 $s\ll c$ ，由计算可得知，普通卷积的计算量大约为 Ghost 模块计算量的 s 倍。GhostBottleNeck 中初始的 Ghost 模块旨在扩充通道数量，而随后的 Ghost 模块则负责将通道数量降低，确保与初始输入的通道数匹配。GhostBottleNeck 模块如图 3-4 所示。将 C3 模块与 GhostBottleNeck 结合，得到 C3Ghost 模块，降低了模型的参数量与计算量。C3Ghost 模块如图 3-5 所示。

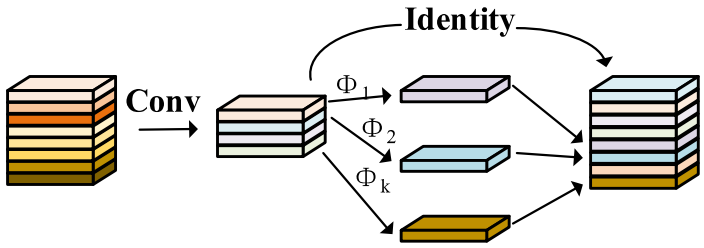


图 3-3 Ghost 模块

Fig. 3-3 Ghost module

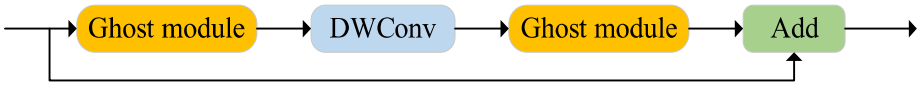


图 3-4 GhostBottleNeck 模块

Fig.3-4. GhostBottleNeck Module

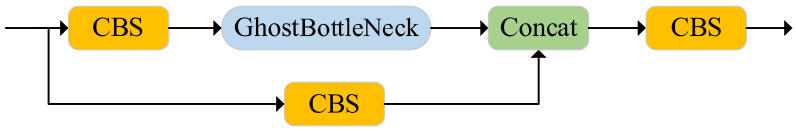


图 3-5 C3Ghost 模块

Fig. 3-5 C3Ghost Module

3.3.2 CA 注意力机制的引入

注意力机制在复杂背景下可以帮助模型忽略与目标物体无关的噪声信息，减少干扰，提高检测的准确性。可以帮助模型聚焦于图像中与目标物体最相关的区域，

从而增强这些区域特征的表达能力，更好地识别和定位目标物体。在工业生产和交通作业，周围环境复杂多变，为了使网络更能关注到安全帽与反光衣，忽略背景信息，引入了注意力机制。

CA（Coordinate Attention）注意力机制是一种轻量化的注意力机制模块，目的是提高卷积神经网络的性能和效率。这种机制不仅考虑了通道间的关系，而且注意了位置信息，这有助于模型更好地定位和识别目标。

CA 注意力机制的显著优点在于它能够整合通道和空间位置信息[45]。SE 注意力忽略了位置信息的关键性，这一信息这对于图像任务至关重要。CA 既能提取通道间的信息，还考虑了横向和纵向的信息，有助于模型关注到目标[46]。CA 注意力机制因其灵活性和轻量级特性，能够轻松地被整合到卷积神经网络的核心位置中，而且能带来性能上的提升[47]。

CA 注意力机制的实现包括两个主要步骤：在通道特征提取中，通过全局平均池化操作，使用两个 1 维向量的编码操作来捕捉单方向上的长距离关系，同时保留另一个方向上的空间信息。在通道注意力计算中，利用提取的位置信息，通过卷积和激活函数处理，生成最终的注意力权重。CA 注意力机制网络结构如图 3-6 所示

具体过程为，输入特征图 X，使用 $(H,1)$ 和 $(1,W)$ 两个池化核，沿着宽度和高度两个方向进行平均池化。表达式如下：

$$z_c^h(h)=\frac{1}{W}\sum_{0\le i\le W}x_c(h,i) \tag{3.4}$$

$$z_c^w(w)=\frac{1}{H}\sum_{0\le i\le H}x_c(j,w) \tag{3.5}$$

其中，c 表示通道，h 表示高度，w 表示宽度。接下来将两个特征图拼接并且进行卷积变换，表达式如下：

$$\mathrm{f}=\partial(F_1([z^h,z^w])) \tag{3.6}$$

$$g^h=\sigma(F_h(\mathrm{f}^h)) \tag{3.7}$$

$$g^w=\sigma(F_w(f^w)) \tag{3.8}$$

$$y_c(i,j)=x_c(i,j)\times g_c^h(i)\times g_c^w(j) \tag{3.9}$$

其中，$[\cdot,\cdot]$ 表示沿空间维度的拼接，F_1 表示 1×1 卷积函数，∂ 表示非线性激活函数，f 表示中间特征图，σ 为 sigmoid 函数，g 表示注意力权重。将 f 沿着为垂直和水平方向的分解为 f^h 和 f^w，然后经过卷积变换，最终得到输出 y。

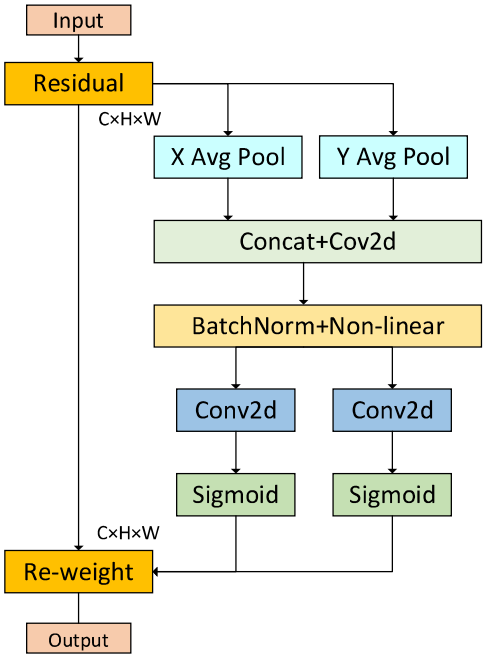


图 3-6 CA 注意力机制网络结构

Fig. 3-6 CA Attention mechanism network structure

3.3.3 WIOU 损失函数的替换

原始的 YOLOv5s 模型采用了 CIoU 损失函数作为其边界框的损失函数[48]。CIoU 损失函数在 IoU 的基础上考虑了预测框与真实框之间的中心点距离以及宽高比，从而能够更全面地评估两个边界框之间的差异。CIoU 损失函数的计算公式如下：

$$CIoU = IoU - \frac{p^2(b,b^{gt})}{c^2} - \alpha v \tag{3.10}$$

其中，IoU 是预测框与真实框的交并比；b 和 b^{gt} 是预测框和真实框的中心点坐标；p 是计算两点之间距离函数；c 为最小包围框对角线长度。

$$\alpha = \frac{v}{(1-IoU)+v} \tag{3.11}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}) \tag{3.12}$$

其中，w^{gt} 和 h^{gt} 分别表示真实框的宽和高；w 和 h 分别表示预测框的宽和高。CIoU 损失函数考虑了预测框与真实框的重叠，位置关系和形状差异，这使得在训练目标检测模型时能够更有效地计算边界框损失函数。但是由于数据集中存在这一些低质量图片，使用 CIoU 损失函数会加剧对这些图片的惩罚力度，导致算法的泛化能力降低。因此本章使用 WIoU 损失函数来替换 CIoU 损失函数[49]。WIoUv1 计算公式如下：

$$Loss_{WIoU} = R_{WIoU} Loss_{IoU} \tag{3.13}$$

$$R_{WIoU}=\exp(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)}) \tag{3.14}$$

其中，$R_{WIoU}\in[1,e)$，会放大正常质量锚框的$Loss_{IoU}$；x和y为预测框的中心点坐标；x_{gt}和y_{gt}为真实框中心点坐标；H_g和W_g为最小包围框的高度和宽度如图 3-7 所示。

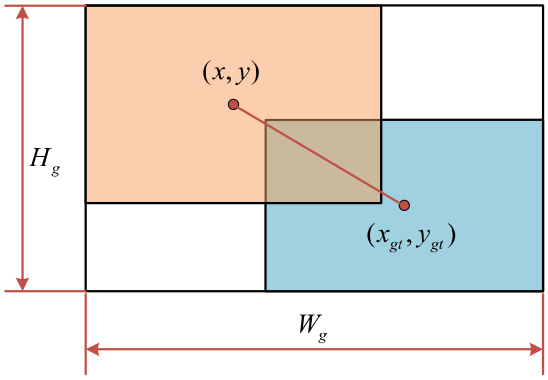


图 3-7 WIoU 示意图

Fig. 3-7 Schematic of WIoU

3.3.4 轻量化 YOLOv5s 网络模型

改进后的轻量化 YOLOv5s 网络模型如图 3-8 所示。其中，使用 GhostConv 替换了网络中原来所有的卷积操作，使用 C3Ghost 模块替换了原本的 C3 模块，极大的降低了模型复杂度，减小了计算量。但是轻量化的网络结构也带来了检测精度的损失。在整个网络中加入了 6 个 CA 注意力机制模块，在不增加过多参数量的情况下，增强对了对感兴趣特征图区域的定位与识别[39]。最后，将 CIoU 损失函数替换为 WIoU 损失函数，提高了模型的泛化能力。

3.4 实验结果与分析

3.4.1 实验环境

实验所使用服务器的操作系统为 Windows10，GPU 为 NVIDIA Quadro RTX3090，CPU 为 Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz。搭建的环境为 python3.11.5，pytorch2.1.0，CUDA12.3。其中训练图片大小为 640×640，每次训练的批量大小设置为 32，训练迭代设置 300 个 epoch，学习率设置为 0.01，其他的参数设置如表 3-1 所示。

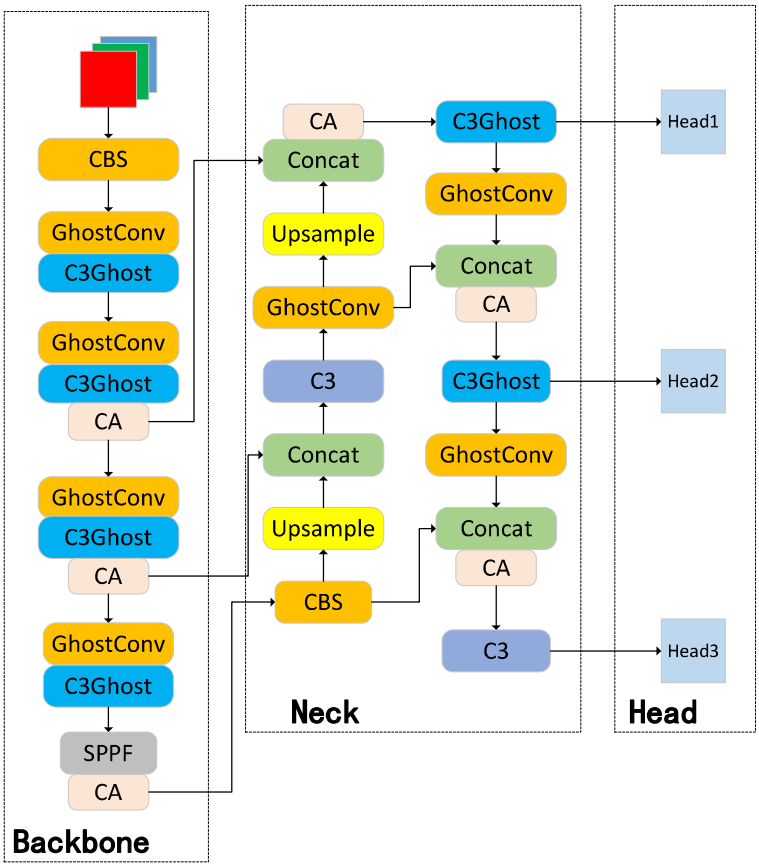


图 3-8 改进的 YOLOv5s 网络结构

Fig. 3-8 The improved YOLOv5s network structure

表 3-1 YOLOv5s 参数设置

Tab. 3-1 YOLOv5s parameter setting

参数	数值
图片大小	640×640
迭代次数	300
批次大小	32
学习率	0.01
动量	0.937
权重衰减	0.0005

3.4.2 数据集

本文使用自制数据集，用 LabelImg 工具对其进行了数据标注与分类，并且剔除了其中不相关的图片。总共分为 4 种类别，分别为 helmet（安全帽），nohelmet（无

安全帽），reflective_clothes（反光衣）和 other_clothes（其他衣服）。数据集图片如图 3-9 所示，其中包含了四种检测类别与数据增强后的图片。安全帽与反光衣初始图片 1083 张，为了增加样本数量，对图像进行了随机旋转，镜像和加噪等操作将图片扩充为 5415 张。将图片按照 8：2 的比例随机划分为训练集与测试集，其中训练集 4332 张图片，测试集 1083 张图片。

由于单张图片中存在着多个标签，为了解各类标签数量，制作了标签统计图如图 3-10 所示。其中反光衣标签数量最少，数量约 4 千；其他衣服标签最多，数量超过 8 千，总计超过 24500 个标签。

图 3-9 数据集图片

Fig. 3-9 Dataset picture

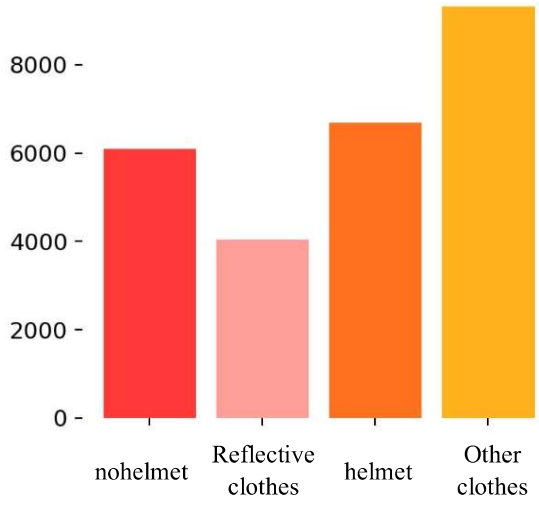
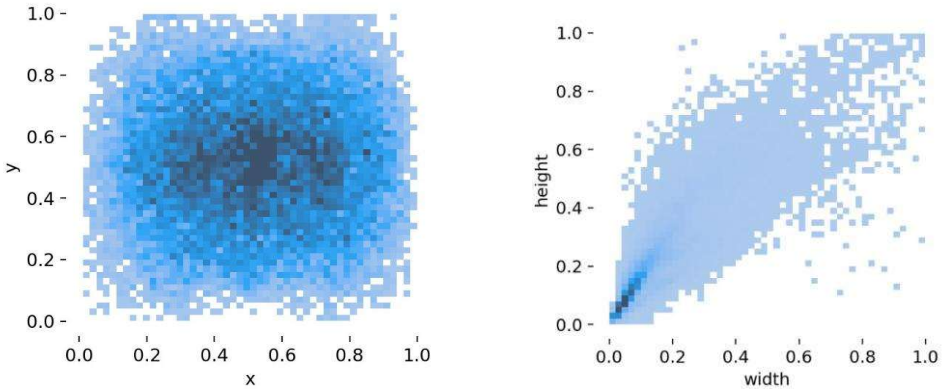


图 3-10 标签数量统计图

Fig. 3-10 Label quantity statistics

为了掌握标注框的分布情况，制作了标注框分布图，如图 3-11 所示。所有标注

框中心点相对于整幅图的位置如图 3-11（a）所示，可见绝大部分目标都集中在图片的中间位置。所有标注框相对于整幅图的高宽比例如图 3-11（b）所示，样本集中分布于左下角，说明数据集中小目标居多。



(a) 中心点分布图　　(b) 高宽比例

图 3-11 标注框分布图

Fig. 3-11 Label box distribution diagram

3.4.3 评估指标

mAP（mean Average Precision）是衡量计算机视觉任务与目标检测任务中的一种常见指标。它是衡量目标类别检测平均精度的重要标准。其中 mAP 的计算公式如下：

$$P = \frac{TP}{TP + FP} \tag{3.15}$$

$$R = \frac{TP}{TP + FN} \tag{3.16}$$

$$AP = \int_0^1 P(R) \cdot dR \tag{3.17}$$

$$mAP = \frac{\sum_{i=0}^{n} AP_i}{N} \tag{3.18}$$

其中，P（Precision）表示精确率，R（Recall）表示召回率，i 表示类别数。TP 表示正样本被预测为正类，FP 表示负样本被预测为正类，FN 表示正样本被预测为负类。mAP@0.5 表示 IOU 阈值为 0.5 的 mAP 值，mAP@0.5:0.95 表示 IOU 阈值为 0.5~0.95，且步长为 0.05 的平均 mAP 值。mAP 值越高，说明模型在检测不同类别对象时的性能越好。这个指标在评估和比较目标检测算法的性能时非常重要。

在目标检测任务中，参数量（Params）和计算量（FLOPs）是评估模型复杂度的重要指标。其中，参数量用来衡量算法的复杂程度。在 YOLO 模型中，参数量取决

于模型的架构。每个卷积层的参数量是由卷积核的数量、大小以及输入和输出的特征图的通道数决定的。卷积的参数量计算公式如下：

$$Params = k^2 \times C_{in} \times C_{out} \tag{3.19}$$

计算量通常用来衡量模型在进行前向传播时所需的浮点运算次数。FLOPs 是“billion floating-point operations”的缩写，即每秒十亿次浮点运算。计算量取决于模特征图的大小，输入和输出通道数的多少。

在 YOLO 模型中，计算量主要来自于卷积层，因为卷积操作是计算密集型的。每个卷积层的计算量可以通过以下公式近似计算：

$$FLOPs = n \times C_{out} \times H_{out} \times W_{out} \times C_{in} \times k^2 \tag{3.20}$$

其中，C_{in} 和 C_{out} 分别表示输入与输出通道数，H_{out} 和 W_{out} 分别表示输出特征图的高和宽，k 表示卷积核大小，将所有层的计算量相加，得到整个模型的计算量。1GFLOPs=10^9FLOPs。

对于具体的 YOLO 的参数量和计算量取决于它们的架构设计，如卷积层的数量、深度、宽度以及输入图像的分辨率。而且，YOLOv5 有不同大小的版本，它们的参数量和计算量随着模型大小的增加而增加。

为了获取特定 YOLO 模型的参数量和计算量，可以使用 PyTorch 的`torchinfo`或`thop`，这些工具可以分析模型的结构并计算其参数量和计算量。需要注意的是，参数量和计算量并不总是直接关联的。一些模型可能有大量的参数，但计算量并不高，这可能是因为模型中有大量的稀疏连接或参数共享。同样，一些参数量少的模型，其计算量可能很高。

3.4.4 实验结果

改进的 YOLOv5s 不同类别 AP 如表 3-2 所示，所有类别的平均精确率为 96.1%，召回率为 90.2%，mAP@0.5 值为 94.8%。可以看出无安全帽的 mAP@0.5 值最小，达到 90.8%，反光衣的 mAP@0.5 值最高，达到 97.5%。反光衣的检测效果最好。改进 YOLOv5 的 mAP@0.5 曲线随着迭代次数增长曲线如图 3-12 所示。50 个 Epoch 时 mAP@0.5 值到达 80%，110 个 Epoch 时 mAP@0.5 值到达 90%，之后开始收敛，最终的 mAP@0.5 值为 94.8%。

P-R 曲线代表精确率与召回率之间的关系。横坐标为召回率，纵坐标为精确率。P-R 曲线与坐标轴围成的面积为此种类型的 AP，所以曲线越靠近右上方，AP 越大，算法性能越好[39]。如图 3-13 所示四条细线分别代表安全帽、无安全帽、反光衣和其

他类衣服，蓝色实线为三种类别的平均精度 mAP@0.5。可以看出，所有曲线都非常靠近右上角，围成面积占到 90%以上。

表 3-2 改进的 YOLOv5s 不同类别 AP

Tab. 3-2 Improved YOLOv5s APs of different classes

类别	精确率/%	召回率/%	mAP@0.5
All	96.1	90.2	94.8
Nohelmet	96.4	88.5	90.8
Reflective_clothes	97.2	93.5	97.5
Helmet	97.2	91.6	95.7
Other_clothes	95.8	90.6	95.1

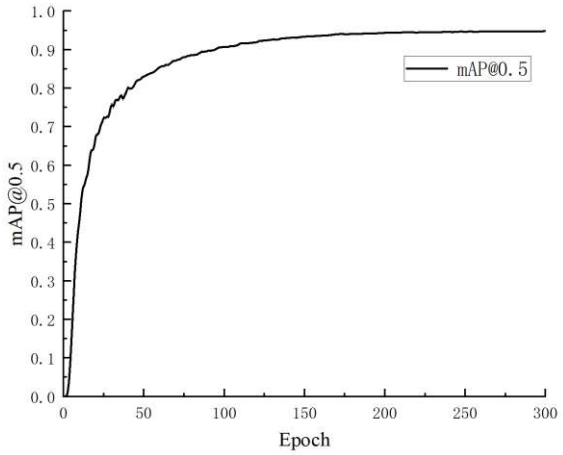


图 3-12 改进 YOLOv5 mAP 曲线

Fig. 3-12 Improved YOLOv5s mAP Curve

3.4.5 消融实验

为了验证本文所提出模型算法有效性，在自制数据集上进行了消融实验如表 3-3 所示。Ghost 模块替换了原有的普通卷积与 C3 模块，极大降低了的模型的参数量，计算量和模型大小。为模型轻量化的主要功能模块。但是模型的轻量化导致了模型检测精度的精度，使模型的检测精度降低到了 93.6%。CA 注意力模块，在不增加模型复杂度的情况下极大的提高了模型的平均精度，达到 95%。与原始模型相比较，mAP@0.5 提高了 0.9%。与替换了 Ghost 的 YOLOv5s 相比较，模型的 mAP@0.5 值提高了 1.4%。WIoU 损失函数的引入也给模型带来了检测精度的提高，与原始模型相比较，mAP@0.5 提高了 0.3%。将各个模块都加入到原始的 YOLOv5s 模型中，不

仅模型的 mAP@0.5 提高了 0.7%，而且模型的参数量降低 2.5×10^6，达到 4.5×10^6。模型的计算量和模型大小分别降低了 6.6 GFLOPs 和 4.6 MB，分别达到了 9.2 GFLOPs 和 9.1MB。

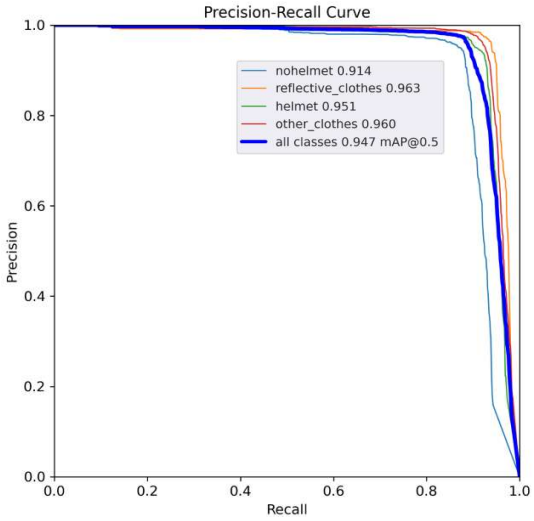


图 3-13 改进 YOLOv5 mAP 曲线

Fig.3-13 Improved YOLOv5s mAP Curve

表 3-3 消融实验

Tab. 3-3 Ablation experiment

模型	mAP@5/%	参数量 /10^6	计算量 /GFLOPs	模型大小/MB
YOLOv5s	94.1%	7.02	15.8	13.7MB
+Ghost	93.6%	4.5	9.1	9.0MB
+CA	95%	7.09	15.9	13.8MB
+WIOU	94.4%	7.02	15.8	13.7MB
+Ghost+CA	94%	4.5	9.2	9.1MB
+CA+ WIOU	95.4%	7.0	15.9	13.8MB
+Ghost +WIOU	93.9%	4.5	9.1	9.0MB
+Ghost+ CA+ WIOU	94.8%	4.5	9.2	9.1MB

3.4.6 对比实验

为了直观地展示检测效果，制作了图 3-14 的检测结果对比。上图为原始的 YOLOv5s 检测结果，下图为改进的 YOLOv5s 检测结果。从图中可以清晰地看出，

四种类别安全帽，无安全帽，反光衣和其他衣服，改进后 YOLOv5s 识别的置信度水平都要高于原始的 YOLOv5s 检测算法。而且，在第二幅图中，原始的 YOLOv5s 存在误检，将背景识别为了其他衣服，而改进后 YOLOv5s 则不存在这种问题。

图 3-14 检测结果对比

Fig. 3-14 Comparison of test results

为了展示改进模型与其他同类型模型的优势，进行了以下表 3-4 的对比实验。可以看出 YOLOv7-Tiny 与原始的 YOLOv5s 算法相比较，模型的复杂程度相差不大，但是 YOLOv7-Tiny 的 mAP@0.5 为 92.7%，比原始 YOLOv5 低 1.4%。改进的 YOLOv5s 的模型复杂度要远低于 YOLOv7-Tiny，而且 mAP@0.5 高了 YOLOv7-Tiny 算法 2.1%。YOLOv7 的检测精度要比改进的 YOLOv5s 高，但其模型复杂度过高，参数量达到 36.5×10^6，计算量达到 103.2 GFLOPs，模型大小达到 71.3 MB。YOLOv3-Tiny 为轻量型的算法，其模型的复杂度比原始 YOLOv5s 略高，且检测精度比原始 YOLOv5s 低 7.7%。YOLOv3 的 mAP@0.5 低于改进的 YOLOv5s 模型 0.2%，且模型参数，计算量远远高于改进后的 YOLOv5s。

表 3-4 对比实验

Tab. 3-4 Contrast experiment

模型	mAP@0.5/%	参数量/10^6	计算量/GFLOPs	模型大小/MB
YOLOv5s	94.1	7.02	15.8	13.7
YOLOv3-Tiny	86.4	8.67	13.0	16.6
YOLOv7-Tiny	92.7	6.01	13.0	11.7
YOLOv3	94.6	61.5	154.6	117
YOLOv7	95.5	36.5	103.2	71.3
改进的 YOLOv5s	94.8	4.5	9.2	9.1

3.5 本章小结

本章首先介绍了 YOLOv5s 算法网络模型的构成与功能。然后对 YOLOv5s 检测算法进行了轻量化的改进，介绍了各个改进模块的原理和主要功能，通过加入 Ghost 模块来降低模型的复杂度，注意力机制模块的添加提高了网络对感兴趣区域的关注，更优秀的损失函数计算方法提高了检测精度。最后，介绍了实验的环境和超参数，实验环境和评估指标。并且在数据集上进行了实验，结果表明改进后的 YOLOv5s 参数量降低了 35.7%，计算量减低了 41.7%，模型大小降低了 33.6%，并且模型的精度提高了 0.7%，达到 94.8%。实验效果良好。