

基于 Yolov4-tiny 算法优化的图像传感系统研发

曾广淼, 王荣杰*

(集美大学 轮机工程学院, 福建 厦门 361021)

摘要: “海洋强国”战略和 5G 技术使我国船舶智能航行技术的发展迈入了崭新阶段。本项目对船舶重叠目标图像传感系统进行优化, 以 Yolov4-tiny 算法为基础, 提出改进的马赛克数据增强方法, 研究针对船舶目标重叠的新型图像传感方法, 对船舶目标重叠的识别效果有一定的提升。其应用的 Yolov4-tiny 小型算法可以灵活装配于小型移动设备, 实现在离线状态下对海面船舶目标的实时监测。

关键词: 马赛克图像增强; Yolov4; 小型移动平台

Image sensing system development based on Yolov4-tiny algorithm optimization

Zeng Guangmiao, Wang Rongjie*

(School of Marine Engineering Jimei University Xiamen 361021)

Abstract: This paper presents a method of license plate character recognition based on the combination of Zernike moment and wavelet transformation features.

Keywords: license plate character recognition; Zernike moment

1 传感器设计背景和应用价值

设计背景:

小型无人艇在离线状态下对周围目标进行识别, 在目标处于重叠遮挡的情况下, 依然可以利用轻量级算法对其进行准确识别。

应用价值: 无人水面巡航艇、无人垃圾清理艇等在近海与内河环境下, 增强图像传感器对水面目标的识别准确率, 提高其对水面环境的感知能力。

2 创新点与优势

(1) 船舶存在处于狭窄水域航行的情况, 此时周围船舶较多, 在观测视野中极易出现相互遮

挡的现象，因此，快速准确地将不同重叠程度目标进行识别定位十分重要。本项目提供一种针对船舶目标重叠的新型图像传感方法与系统，对训练数据集中的图像进行数据增强，在不改变网络结构的情况下，加强目标识别算法对船舶重叠目标的识别能力。

(2) 针对处于离线状态下的小型移动平台，选择合适的轻量级算法，在保持其识别速度的情况下提升识别准确率，降低不同分辨率的输入图像对算法识别性能的影响。

(3) 通过在船舶数据集不同的海域进行实时检测测试，结果表明，经过改进马赛克方法训练后的算法在船舶重叠目标的检测当中准确率更高，证明了改进方法的有效性和鲁棒性。

3 实现方案简介

3.1 设计原理

Yolo[1]算法自 2016 年首次提出以来受到了广泛的关注，而随后提出的 YOLOv2[2]和 YOLOv3[3]算法在其基础上对模型结构、数据预处理方式、损失函数计算方法进行了一系列的改进，极大提升了目标检测的速度与准确率。而 YOLOv4[4]算法在 YOLOv3 算法的计算方式上，增添了很多优化技巧，在识别速度不变的情况下提升了识别准确率。

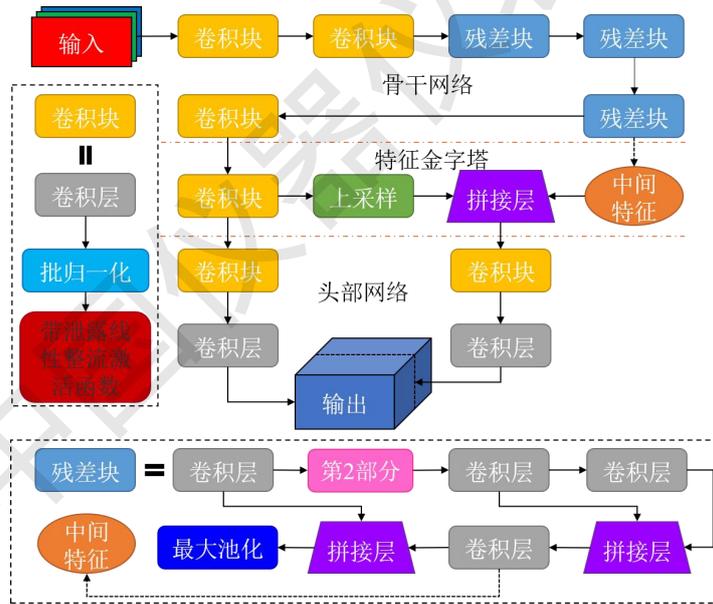


图 1 YOLOv4-tiny 网络的特征结构

YOLOv4-tiny 网络是基于 YOLOv4 网络进行简化，在牺牲一部分识别准确率的情况下，减少了 10 倍的参数量。从 YOLOv4 网络中约 6000 万的参数降至 YOLOv4-tiny 中约 600 万的参数。其网络结构图如图 1 所示。

其中，骨干网络中的卷积块由卷积层、批归一化层[5]、带泄露线性整流[6]激活函数组成。而残差块是融合了跨阶段局部网络[7]思想(CSPNet)的 CSPDarknet53-Tiny，其结构如图

2 所示。

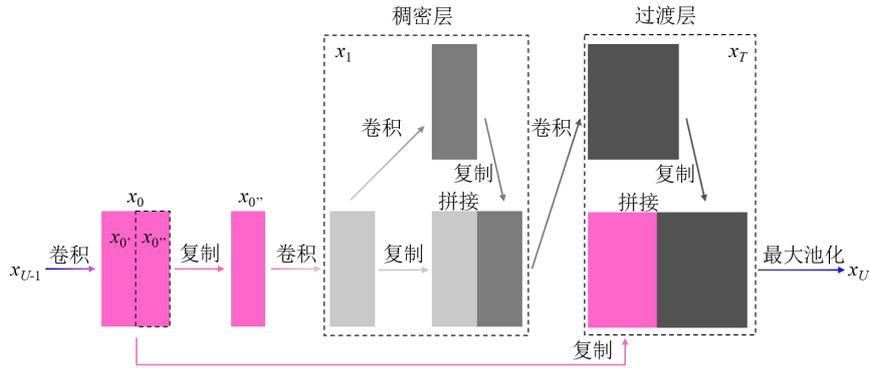


图 2 CSPDarknet53-Tiny 网络中残差块的结构示意图

它由稠密层和连接层组成，首先对上一卷积层的输出 x_{U-1} 进行卷积操作，生成新的卷积层，将其输出 $x_0=[x_0', x_0'']$ 分成前后两个部分 x_0' 和 x_0'' 进行前项计算，在 Yolov4-tiny 的网络结构中，先取第二部分进行前项传播，再将第一部分与第二部分一起直接连接到阶段的末尾，跳过稠密层，等到第 2 部分的 x_0'' 完成前项计算之后在过渡层中与 x_0 进行特征拼接，得到输出 x_T ，其经过最大池化后产生残差块的输出 x_U 。残差块前向计算与反向传播的过程如式(1)- (2)所示。

$$\begin{aligned} x_T &= w_T \cdot [x_0', x_1] \\ x_U &= w_U \cdot [x_0, x_T] \end{aligned} \quad (1)$$

$$\begin{aligned} \omega'_T &= f_T(\omega_T \cdot \{g_0, g_1\}) \\ \omega'_U &= f_U(\omega_U \cdot \{g_0, g_T\}) \end{aligned} \quad (2)$$

其中， ω_i 和 ω'_i 是前向计算与反向传播时的权重， f_i 表示权重更新的函数， g_i 表示传播到第 i 层的梯度， i 等于 T 或 U ，分别代表着连接层或残差块的输出。所以，利用跨阶段局部网络的结构在反向传播时，可以将不同通道上的梯度分别进行积分，比如梯度信息经过稠密层时，只会改变 x_0'' 通道上的权重却不会影响到 x_0' 。从而在保留不同深度的特征值的同时，减少了过多的重复梯度信息，在不影响网络特征提取效果的情况下，减小了内存的开销，提高了网络的运算速度。

在骨干网络之后，利用特征金字塔结构[8]对网络特征进行优化，Yolov4-tiny 网络中小型特征金字塔的实现方式如图 3 所示。

中间特征是骨干网络的第三个残差块中第四个卷积层的输出，与网络中第四个卷积块的输出经过上采样后的特征进行拼接。在图 1 可以看出，骨干网络只包含前三个 CBL 层，因此图 3 中进行拼接操作生成的输出代表了浅层网络特征和深层网络特征的融合。由于经过了

多层卷积自上而下的特征提取，深层网络会保留绝大部分大目标的特征值，小目标的特征值被保存下来的很少甚至为零。因此采用特征金字塔结构提取多个不同层级网络的特征，通过上采样放大之后，将其自下而上的拼接在一起，实现了多个层级的特征融合，提高了网络对多种分辨率下不同大小目标的识别能力。

之后将小型特征金字塔的两部分输出接入头部网络进行计算，生成两组包含不同感受野的图像，分别对自身包含的先验框进行调整，利用非极大值抑制的方法，针对原图中大小不同的目标进行识别与检测，提高神经网络对多尺度目标的整体检测能力。

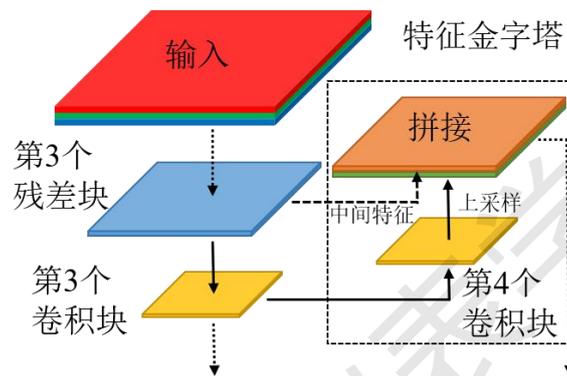


图3 小型特征金字塔的结构示意图

3.2 设计方法

Mosaic 方法是在切割混合^[9]方法上进行拓展，生成的一个新的数据增强算法，不同于切割混合方法的两张图片覆盖融合，它而是利用四张图片进行剪裁拼接，形成一张新的图片。这种方法可以更好地丰富目标物的背景，防止由于训练集背景相似而导致的网络泛化能力降低。

Yolov4-tiny 算法的输出图像中包含有两种不同的感受野，而 Yolov4 算法的输出图像中则有三种不同的感受野，因此 Yolov4-tiny 算法对于多尺度目标的识别能力就会相对弱一些。所以，对数据增强方法进行改进，增强网络的泛化能力就显得尤为重要。

改进的马赛克数据增强方法如图 4 所示。原始 Mosaic 方法是以图 2 中的上中两条橙色通道进行的特征增强，而改进 Mosaic 方法在原先的基础上增加了下方的金色通道，采用三条通道进行特征增强。第三条金色通道的输出，相较于上面两种，是对每行每列排列的图像数进行了增加而得到的。为了方便说明，将以 3*3 的规格新生成九合一的图片称之为 m9，将以 2*2 的规格生成的四合一图片称之为 m4，将不经过合并以 1*1 规格生成的图片称之为 m1，m1、m4 和 m9 的比例为 o:p:q。这样的组合方式，一定程度上使得训练数据集的尺度变化特性更加多样，从而进一步丰富了数据集，增加了背景的复杂程度。因此，网络可以在

复杂背景的干扰下，更专注于对目标物特征的提取，增强了网络的鲁棒性。

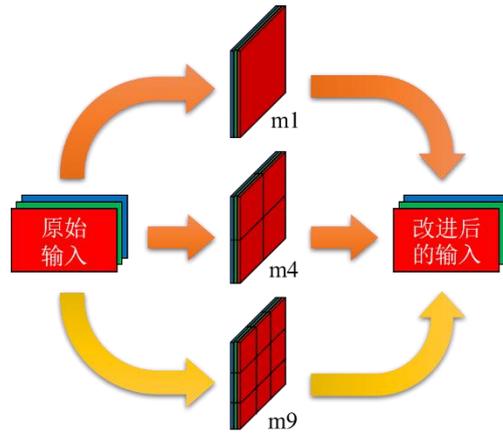


图 4 改进的马赛克数据增强方法

九合一图像 m_9 的生成方式如图 5 所示，主要分为 A, B, C 三个阶段。在 A 阶段，以输入图像的宽和高(W, H)作为边界值，首先对图像进行缩放，X 轴和 Y 轴的缩放倍率为 t_X 和 t_Y ，如式(3)和式(4)所示。

$$t_X = f_{rand}(t_W, t_W + \Delta t_W) \quad (3)$$

$$t_Y = f_{rand}(t_H, t_H + \Delta t_H) \quad (4)$$

其中， t_W 和 t_H 分别为宽和高缩放倍率的最小值， Δt_W 和 Δt_H 分别为宽和高缩放倍率随机区间的长度，均为超参数。 $f_{rand}()$ 表示随机值函数。

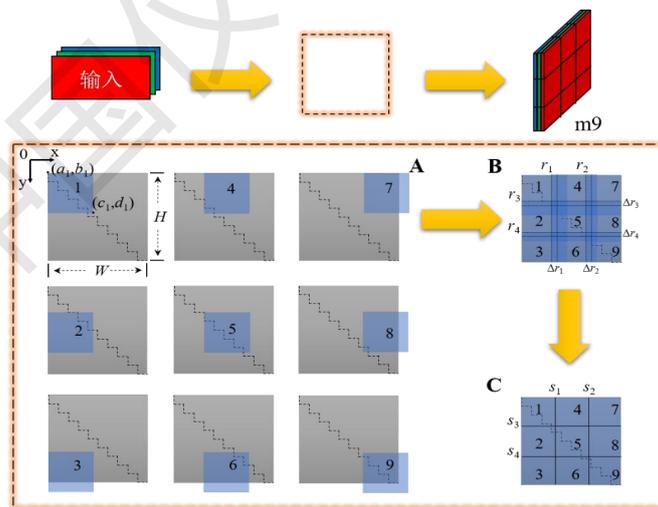


图 5 m_9 图像的生成流程图

图像缩放后的左上角和右下角的坐标为 $[(a_i, b_i), (c_i, d_i)]$ ，由式(5)到式(8)可得。

$$a_i = \begin{cases} 0 & i=1,2,3 \\ W \cdot r_1 & i=4,5,6 \\ W \cdot r_2 & i=7,8,9 \end{cases} \quad (5)$$

$$b_i = \begin{cases} 0 & i=1,4,7 \\ H \cdot r_3 & i=2,5,8 \\ H \cdot r_4 & i=3,6,9 \end{cases} \quad (6)$$

$$c_i = a_i + W \cdot t_W \quad (7)$$

$$d_i = b_i + H \cdot t_H \quad (8)$$

其中, r_1 和 r_2 分别为 X 轴的除 0 点以外的两组图像左上角坐标点与 0 点的距离占总宽度的比值, r_3 和 r_4 分别为 Y 轴的除 0 点以外的两组图像左上角坐标点与 0 点的距离占总高度的比值, 同样也均为超参数。而灰色区域中的黑色短划线为比例尺, 每一小段代表宽或高的十分之一, 利用比例尺可以看出, 第 2 张至第 9 张的图像和第 1 张的缩放比例一致, 宽和高都是原来的 t_W 和 t_H 倍。

在 B 阶段, 需要将上一阶段裁剪好的 9 张图片进行拼接, 并裁减掉溢出边界框的部分, 可以看到合并后的图像存在一定程度的重叠, 因此需要对每个小区域进行划分。从 A 阶段的示意图中可知, 当缩放后的图片按照坐标在指定位置进行放置时, 会存在溢出边框的情况。此时需对溢出的部分进行裁剪, 如式(9)-式(10)所示。

$$c'_i = \begin{cases} c_i & \text{if } c_i < W \\ W & \text{if } c_i \geq W \end{cases} \quad (9)$$

$$d'_i = \begin{cases} d_i & \text{if } d_i < H \\ H & \text{if } d_i \geq H \end{cases} \quad (10)$$

在边缘裁剪之后, 利用 8 条两两平行的圆点线围成的 4 个方形区域, 将其作为分割线的随机区间。其中, $r_i=(r_1, r_2, r_3, r_4)$ 的值等于分割线坐标与 0 点的距离占边界长度的比值, Δ_i 为分割线随机区间的长度。

在 C 阶段, 将对内部重叠部分进行第二次裁剪, 其分割线坐标 s_i 可由式(11)得

$$s_i = f_{rand}(r_i, r_i + \Delta r_i) \quad i=1,2,3,4 \quad (11)$$

经过裁剪以后, 得到了拼接完成的 m_9 图像。由于原图在缩放拼接过程中存在部分缺失, 对于处在原图边缘的目标有可能在操作过程中被截取部分或者完全截掉, 因此, 同样需要对这些目标对应的真实框进行裁剪甚至剔除, 以满足目标检测的需要。

生成 m4 图像的方法与生成 m9 类似，如图 6 所示。左上角坐标(a*i, b*i)与分割线坐标 s*i 由式(12)-(14)所示。

$$a_i^* = \begin{cases} 0 & i=1,2 \\ W \cdot r_1^* & i=3,4 \end{cases} \quad (12)$$

$$b_i^* = \begin{cases} 0 & i=1,3 \\ H \cdot r_2^* & i=2,4 \end{cases} \quad (13)$$

$$s_i^* = f_{rand}(r_i^*, r_i^* + \Delta r_i^*) \quad i=1,2 \quad (14)$$

其中，以*作为 m4 图像与 m9 图像的区别符号。缩放倍率 t*X、t*Y、 $\Delta t*W$ 和 $\Delta t*H$ 及右下角坐标(c*i, d*i)与 m9 图像的计算方式一致，由于没有出现边框外溢出，所以只需要对内部重合部分进行分割裁剪。

m1 图像由于不存在多张图片拼接的情况，因此进入网络之前只需要通过翻转、色域变化等常规方法进行特征增强。

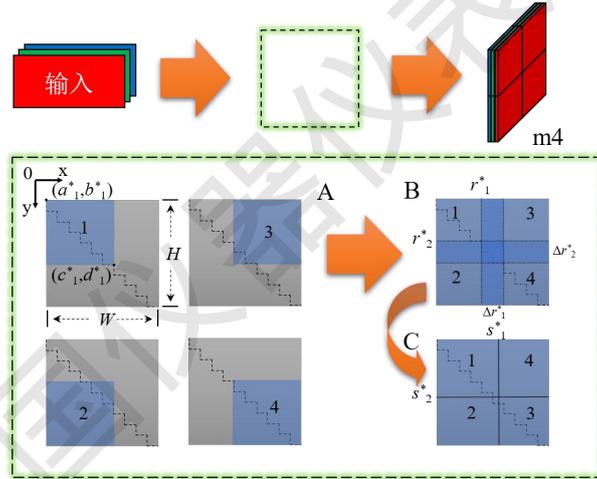


图 6 m4 图像的生成流程图

3.4 实验验证过程

目前广泛使用的数据集，如 VOC 数据集[9]、COCO 数据集[10]等包含的船舶只被分为了 1 类，存在图片尺寸随机且分辨率较低（不超过 640*360）的特点。相比之下，本文中用于训练及测试的船舶数据集[12]一共包含 7000 张图片，其分辨率为 1920*1080，它们截取自监控摄像机所拍摄的视频片段，而这些摄像机是属于沿海岸线部署的海面监控系统，其包括 50 个不同位置的 156 个摄像机。数据集中包含 6 种不同类型的船舶，其数目与类别如表 1 所示。

这些图像中的船舶拥有不同的光照条件、观测角度、远近尺度以及重叠程度，使数据集

的复杂程度大大提升，增加了目标检测算法识别的难度。

表 1 船舶数据集目标物类别与数量

船舶类别	数量/个
矿砂船	2084
散货船	1811
杂货船	1426
集装箱船	898
渔船	1539
客船	455

本文的算法是在开源的神经网络框架 Pytorch (3.8.5)上进行实现的。计算工作站配置包含 1 个 GPU(GeForce RTX 3090), CPU(AMD Ryzen 9 3950x 16 Core/ 3.5 GHz/72 M), 以及 128 G RAM。小型移动测试平台是基于 NVIDIA Jetson Xavier NX 开发板进行搭建, 由 1 个 1080p 摄像头模组, 供电模组, 显示输出模组及控制模组组成。

在训练开始之前, 先将数据集进行分类。选择 6000 张几乎无遮挡的船舶目标图片作为训练集, 1000 张发生不同程度重叠的船舶图片作为测试集。通过对目标识别算法的优化, 使目标船舶在发生不同程度的重叠与遮挡时, 能够更加快速且准确地捕捉到目标, 减少提高识别准确率。实验中 Mosaic 方法的参数如表 2 所示, 网络优化方法中的参数如表 3 所示。

表 2 马赛克数据增强方法中的实验参数

参数	数值	参数	数值	参数	数值
W	608	$\Delta r3$	0.05	$\Delta r*1$	0.2
H	608	$\Delta r4$	0.05	$\Delta r*2$	0.2
r1	0.3	tW	0.4	t*W	0.4
r2	0.65	tH	0.4	t*H	0.4
r3	0.3	ΔtW	0.05	$\Delta t*W$	0.2
r4	0.65	ΔtH	0.05	$\Delta t*H$	0.2
$\Delta r1$	0.05	r*1	0.4		
$\Delta r2$	0.05	r*2	0.4		

表 3 网络优化方法中的参数

参数	数值
φ	0.1
γ_0	1×10^{-3}
ε	0.95

训练数据集中用于训练和验证的图片以 9 比 1 的比例随机分割，训练开始后，网络经过 100 次迭代后停止，图 7 表示 Yolov4-tiny 算法在不同迭代次数下的识别准确率，图中图例的数值为 o:p:q 的值，准确率的高低由各类别平均精度的平均值(mean Average Precision, mAP)来表示。

从图 7 中可以看出，在趋向于平稳的后二十次迭代过程中，黄色曲线的值略高于绿色曲线的值，使用 Mosaic 方法之后 Yolov4-tiny 算法的识别结果略有提升，而蓝色曲线表示的改进 Mosaic 方法则大大提高了识别准确率，甚至高于使用原始 Mosaic 方法的 Yolov4 算法的识别准确率。所以改进 Mosaic 方法不仅提高了 Yolov4-tiny 算法的识别准确率，相对于 Yolov4 算法大大提高了船舶识别的检测速度。运用 m4 和 m9 方法的数据增强实验过程如图 8 所示。

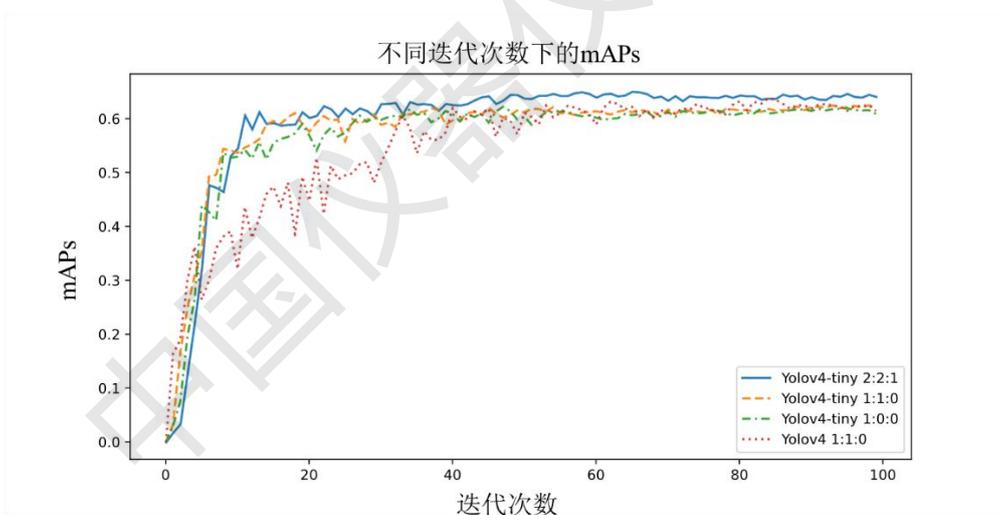


图 7 Yolov4-tiny 算法在不同迭代次数下的识别准确率

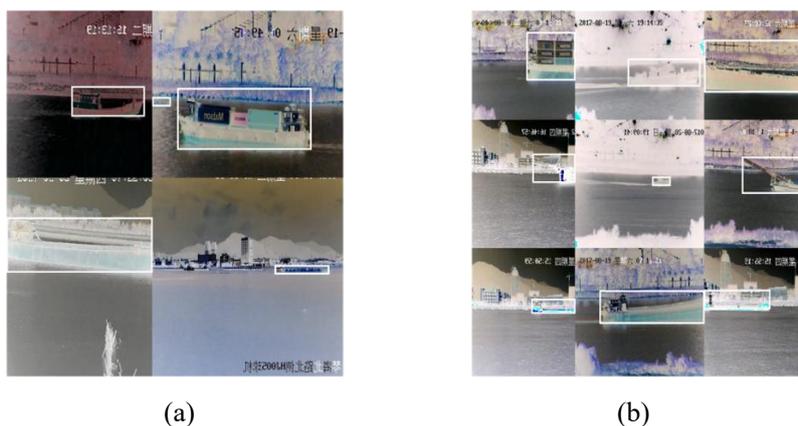


图8 m4与m9方法的实验过程示意图, (a) m4, (b) m9

表4 不同比例下 Mosaic 数据增强方法对目标识别准确率的影响

算法类型	m1,m4,m9 的比例	m1 占 比	m4 占 比	m9 占 比	100 次迭代 中的最高 mAP	后 20 次迭 代中的平 均 mAP	相对于原 始 Mosaic (1:1:0) 的 提升比例
Yolov4-tiny	1:0:0	100%	0%	0%	62.28%	61.54%	99.55%
	0:1:0	0%	100%	0%	60.17%	55.98%	96.18%
	0:0:1	0%	0%	100%	31.34%	30.69%	50.10%
	1:1:0	50%	50%	0%	62.56%	61.99%	100.00%
	1:0:1	50%	0%	50%	60.91%	58.24%	97.36%
	1:1:1	33%	33%	33%	62.39%	60.53%	99.73%
	2:2:1	40%	40%	20%	65.06%	64.09%	104.00%
	2:1:1	50%	25%	25%	63.11%	61.44%	100.88%
	3:2:1	50%	33%	17%	63.78%	62.49%	101.95%
	4:2:1	57%	29%	14%	63.31%	62.86%	101.20%
Yolov4	4:3:2	45%	33%	22%	63.80%	62.21%	101.98%
	5:3:2	50%	30%	20%	65.01%	64.48%	103.92%
Yolov4	1:1:0	50%	50%	0%	63.90%	62.56%	102.14%

表 5 不同数据增强方法的目标识别准确率

算法类型	数据增强方法	训练集进行数据增强的图片比例	100 次迭代中的最高 mAP	后 20 次迭代中的平均 mAP
Yolov4-tiny	Mixup[13]	100%	36.19%	33.04%
		50%	40.16%	38.19%
	Cutout[14]	100%	61.17%	60.24%
		50%	61.75%	61.31%
	Cutmix[15]	100%	22.69%	20.18%
		50%	34.77%	24.19%
	Mosaic	100%	60.17%	55.98%
		50%	62.56%	61.99%
	改进的 Mosaic	60%(40% m_4 +20% m_9)	65.06%	64.09%

图 7 中识别准确率最高的曲线是使用改进 Mosaic 方法进行训练的，其使用 m_1 、 m_4 和 m_9 方法作为输入的概率按照比例 $o:p:q=2:2:1$ 来进行。为了探究比例的不同对识别准确率的影响，本文进行了多组实验比较，结果如表 4 所示。同时，将改进 Mosaic 方法与三种同类代表性数据增强方法进行了比较，结果如表 5 所示。

从表 4 的前 5 行的对比可以看出，如果训练集中的所有图像都使用 m_4 或 m_9 方法进行数据增强，而抛弃原始图像的话，相较于不使用数据增强方法的情况，其识别准确率反而会降低，而无论是 m_4 或 m_9 方法，结合 50% 的原始图像同时训练，其准确率都会提升。因此，即使使用数据增强方法，也不可以丢弃原始图像，这一点同时也可以通过表 5 中的前 8 行数据来印证。

而同比例使用 m_1 、 m_4 及 m_9 三种方法进行训练时（当 $o:p:q=1:1:1$ 时），其准确率较原始 Mosaic 方法（当 $o:p:q=1:1:0$ 时）几乎一致，并没有得到预期的提升，考虑到可能是因为按照此方法训练时原始图片的比例过小（仅有 30%），所以在表 4 的第 7 至 12 行进行了多种比例的实验，并且将 m_1 的占比调至 40% 以上，准确率较原始 mosaic 方法都有了不同程度的提升，这说明数据集应当以 m_1 数据作为重点，让网络学习好目标的整体特征，在此基础上利用 m_4 与 m_9 分别对局部特征进行加强学习，提高网络的泛化能力。并且可以发现，当 m_4 与 m_9 的占比只差在 10% 至 20% 且 m_1 的占比在 40% 至 50% 时，网络的训练效果会更好，即当 $o:p:q=2:2:1$ 和 $o:p:q=5:3:2$ 时，mAP 的最高值相对原始 mosaic 方法提升了 2.5%，

并且高于使用原始 Mosaic 方法训练的 YOLOv4 算法。

为了进一步测试网络对重叠目标的识别能力，利用小型移动测试平台对海面进行实时检测，测试地点位于鼓浪屿(中国，厦门)。以一段在行进过程中发生重叠的两艘渔船的实时视频片段作为实验内容，计算网络在各个时刻的识别能力，视频片段分辨率为 1080P，时长 38 秒，每秒 24 帧，共 912 帧，测试结果如图 9 所示。实测实验中选用的权重文件为 100 次迭代中最高 mAP 所对应的权重。

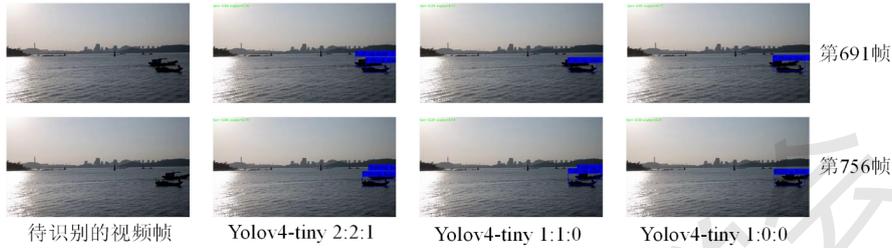


图 9 重叠检测实验对比图

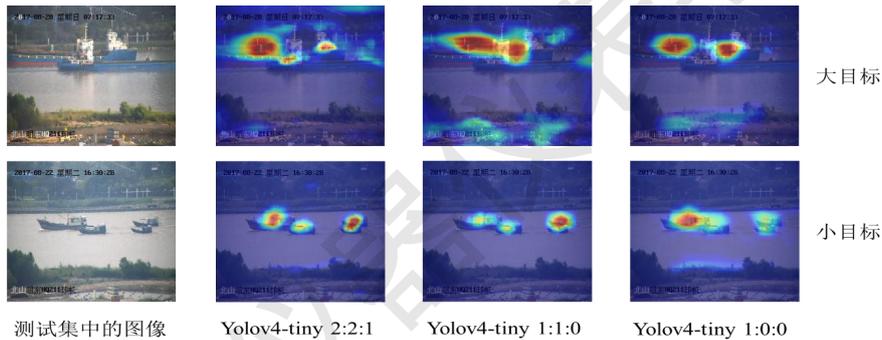


图 10 不同训练方式下梯度加权类激活映射图的对比

在图 9 中可以看出，第 691 帧两艘船已经发生严重重叠时，使用改进 Mosaic 方法的 YOLOv4-tiny 算法依然可以将两艘船同时识别出来，而使用原始 Mosaic 方法和未使用 Mosaic 方法的 YOLOv4-tiny 算法只能识别其中的一艘船。在第 756 帧两艘船正在摆脱重叠的过程中，使用改进 Mosaic 方法和原始 Mosaic 方法的 YOLOv4-tiny 算法已经可以将两艘船同时识别出来，而未使用 Mosaic 方法的 YOLOv4-tiny 算法依然只能识别其中的一艘船。这进一步说明了改进 Mosaic 方法对提升重叠目标检测能力的有效性。

与此同时，为了探究网络对于目标物特征识别能力的影响，其梯度加权类激活映射图 (Grad-CAM) [16]如图 10 所示。

在图 10 中，通过对比在不同训练方式下的梯度加权类激活映射图，可以直观地观察到网络对于测试图片感兴趣区域的不同。第一，对于发生重叠且不完整的船舶大目标，当 $o:p:q=2:2:1$ 时，此时网络所关注的中心点有三个，其中两个位于船舶上层建筑的位置，可以

更好地对船舶的个数进行判断，而且重点关注了图片的上半区域，相比于其他两种情况，对下半区域的背景特征有了更好的抗干扰能力。另外，相较于未使用 Mosaic 方法训练的情况，原始 Mosaic 方法的确增强了网络对于船舶的关注度。第二，对于发生重叠且完整的船舶小目标，当 $o:p:q=2:2:1$ 时，网络对于不同船舶位置的关注点较为准确且边界感较强，同样可以增加对船舶个数的判断能力，而未使用 Mosaic 方法训练下的网络，对于小目标的识别能力明显更弱。因此，对于不同尺度的船舶目标，改进的 Mosaic 方法在一定程度上可以提高网络的识别能力。

由于小型移动设备的算力有限，在 1080p 分辨率下识别速度较慢，因此对不同分辨率视频中的船舶目标进行检测测试，网络的识别速度由表 6 所示。

表 6 不同分辨率下的识别速度对比

网络类型	识别速度/fps				网络权重个数/个	网络权重大小/MB
	1080p	720p	480p	360p		
Yolov4-tiny	6.34	8.68	10.48	11.38	5,885,666	22.45
Yolov4	2.03	2.27	2.41	2.46	63,964,611	244.01

表 7 采用不同 Mosaic 方法在不同分辨率下的对于重叠问题的识别能力对比

视频分辨率	采用的数据增强方法	无法分离	无法分离的帧数/结束的帧	无法分离的帧个数/的帧个数/	无法分离的时间/秒	无法分离的时间相对于原始 Mosaic 方法
1080p	未采用 Mosaic 方法	654	804	151	6.29	196%
	原始 Mosaic 方法	676	752	77	3.21	100%
	改进 Mosaic 方法	692	755	64	2.67	83%
720p	未采用 Mosaic 方法	649	810	162	6.75	210%
	原始 Mosaic 方法	673	752	80	3.33	104%
	改进 Mosaic 方法	691	755	65	2.71	84%
480p	未采用 Mosaic 方法	650	912	263	10.96	342%
	原始 Mosaic 方法	658	753	96	4.00	125%
	改进 Mosaic 方法	692	755	64	2.67	83%
360p	未采用 Mosaic 方法	649	912	264	11.00	343%
	原始 Mosaic 方法	656	754	99	4.13	129%
	改进 Mosaic 方法	692	756	65	2.71	84%

由表 6 可以看出，由于 Yolov4 网络结构较复杂，权重参数较多，导致其识别时需要的算力较高，因此在 4 种分辨率下的识别速度都很低，而 Yolov4-tiny 算法网络结构较简单，权重参数较少，识别时需要的算力较低。Yolov4-tiny 算法在 360p 视频分辨率下的识别速度为其在 1080p 视频分辨率下的 179.50%，为 Yolov4 算法在 360p 视频分辨率下的识别速度的 462.60%。

由于不同分辨率下不同算法的识别速度不同，因此改进 Mosaic 在不同分辨率下的识别能力的重要性便不言而喻。于是，通过对比在重叠开始至重叠结束的区间内，以应用不同 Mosaic 方法的 Yolov4-tiny 算法将两艘船正确识别并分离的帧个数，判断其对于重叠问题的识别能力，具体实验结果如表 7 所示。

从表 7 中可以看出，随着分辨率的下降，各个算法对于重叠目标识别速度都有不同程度的减弱，其减弱的程度可以由无法分离的时间增长率来表示，如图 11 所示。由此可见，使用改进 Mosaic 方法的 Yolov4-tiny 算法在 4 种不同的分辨率下都取得较大优势，不仅在船舶目标产生重叠时，无法分离重叠目标的时间较短，而且在分辨率发生下降时也可以维持高识别率。

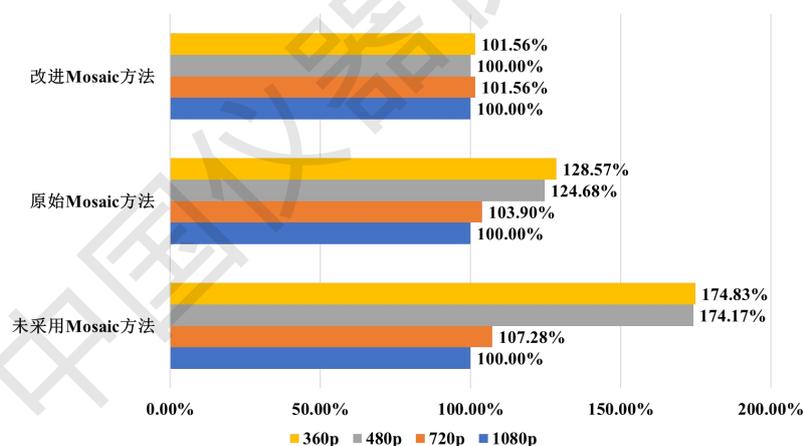


图 11 采用不同 Mosaic 方法在不同分辨率下无法分离的时间增长率

实验证明，改进 Mosaic 数据增强方法对船舶重叠目标的识别效果有一定的提升，应用于 Yolov4-tiny 算法可以部署于小型移动设备，可以灵活地装配在各类平台上，从而实现在离线状态下对海面船舶目标的实时监测。

小型移动测试平台实物图如图 12 所示。



图 12 小型移动测试平台示意图

参考文献:

- [1] Redmon J., Divvala S., Girshick R. You only look once: Unified, real-time object detection[A]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)[C], NV, Las Vegas, USA: IEEE, 2016, 779-788.
- [2] Redmon J., Farhadi A. YOLO9000: Better, faster, stronger[A]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)[C], HI, Honolulu, USA: IEEE, 2017, 6517-6525.
- [3] Redmon J., Farhadi A. YOLOv3: An incremental improvement[EB/OL]. <https://arxiv.org/abs/1804.02767> , 2018.
- [4] Bochkovskiy A., Wang C., Liao H. YOLOv4: Optimal speed and accuracy of object detection[EB/OL]. <https://arxiv.org/abs/2004.10934> , 2020.
- [5] Ioffe S., Szegedy C. Batch Normalization: accelerating deep network training by reducing internal covariate shift[EB/OL]. <https://arxiv.org/abs/1502.03167> , 2015.
- [6] Xu B., Wang N., Chen T. Empirical evaluation of rectified activations in convolutional network[EB/OL]. <https://arxiv.org/abs/1505.00853> , 2015.
- [7] Wang C., Liao M., Wu Y. CSPNet: A new backbone that can enhance learning capability of CNN[A]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)[C], Seattle, WA, USA: IEEE, 2020, 1571-1580.
- [8] Lin T., Dollár P., Girshick R. Feature pyramid networks for object detection[A]. 2017 IEEE

- Conference on Computer Vision and Pattern Recognition (CVPR)[C], HI, Honolulu, USA: IEEE, 2017, 936-944.
- [9] Yun S., Han D., Chun S. CutMix: Regularization strategy to train strong classifiers with localizable features[A]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)[C], Seoul, Korea (South): IEEE, 2019, 6022-6031.
- [10] Everingham M., Van G., Williams C. The pascal visual object classes (VOC) challenge[J]. International journal of computer vision, 2010, 88(2): 303-338.
- [11] Lin T., Maire M., Belongie S. Microsoft coco: Common objects in context[A]. European Conference on Computer Vision[C]. Cham, Switzerland: Springer, 2014, 740-755.
- [12] Shao Z., Wu W., Wang Z. Seaships: A large-scale precisely annotated dataset for ship detection[J]. IEEE transactions on multimedia, 2018, 20(10): 2593-2604.
- [13] Zhang H., Cisse M., Dauphin YN. Mixup: Beyond empirical risk minimization[A]. 2018 International Conference on Learning Representations (ICLR)[C], Vancouver, BC, Canada, 2018.
- [14] DeVries T., Taylo GW. Improved regularization of convolutional neural networks with cutout[EB/OL]. <https://arxiv.org/abs/1708.04552> , 2017.
- [15] Yun S., Han D., Oh SJ. Cutmix: Regularization strategy to train strong classifiers with localizable features[A]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)[C], Seoul, Korea (South): IEEE, 2019, 6023-6032.
- [16] Selvaraju R., Cogswell M., Das A. Grad-CAM: Visual explanations from deep networks via gradient-based localization[A]. 2017 IEEE/CVF International Conference on Computer Vision (ICCV)[C], Venice, Italy: IEEE, 2017, 618-626.