

# 基于改进 Faster R-CNN 的红外舰船目标检测算法

顾佼佼<sup>1</sup>, 李炳臻<sup>2</sup>, 刘克<sup>1</sup>, 姜文志<sup>1</sup>

(1. 海军航空大学 岸防兵学院, 山东 烟台 264001; 2. 中国人民解放军 95668 部队, 云南 昆明 650000)

**摘要:** 针对 Faster R-CNN 算法中对于红外舰船目标特征提取不充分、容易出现重复检测的问题, 提出了一种基于改进 Faster R-CNN 的红外舰船目标检测算法。首先通过在主干网络 VGG-16 中依次引出三段卷积后的 3 个特征图, 将其进行特征拼接形成多尺度特征图, 得到具有更丰富语义信息的特征向量; 其次基于数据集进行 Anchor 的改进, 重新设置 Anchor boxes 的个数与尺寸; 最后优化改进后 Faster R-CNN 的损失函数, 提高检测算法的整体性能。通过对测试数据集进行分析实验, 结果表明改进后的检测算法平均精确度达到 83.98%, 较之于原 Faster R-CNN, 精确度提升了 3.95%。

**关键词:** 深度学习; 目标检测; 舰船目标; 红外图像; Faster R-CNN

中图分类号: TP399

文献标识码: A

文章编号: 1001-8891(2021)02-0170-09

## Infrared Ship Target Detection Algorithm Based on Improved Faster R-CNN

GU Jiaojiao<sup>1</sup>, LI Bingzhen<sup>2</sup>, LIU Ke<sup>1</sup>, JIANG Wenzhi<sup>1</sup>

(1. Naval Aviation University, Coast Guard Academy, Yantai 264001, China;

2. Unit 95668 People's Liberation Army of China, Kunming 650000, China)

**Abstract:** To solve the problem of insufficient feature extraction and repeated detection of infrared ship targets by the Faster R-CNN algorithm, a ship target detection algorithm based on an improved Faster R-CNN is proposed. First, three feature graphs are drawn from the backbone network, VGG-16, after a three-segment convolution, and the features are spliced to form a multi-scale feature graph to obtain a feature vector with richer semantic information; second, the Anchor is improved based on the dataset, and the number and size of the Anchor boxes are reset; finally, the loss function of the improved Faster R-CNN is optimized to improve the feature extraction ability of the target. An analysis of the experimental results on the test dataset demonstrates that the average accuracy of the improved detection algorithm was 83.98%, which is 3.95% higher than that of the original Faster RCNN.

**Key words:** deep learning, target detection, ship target, infrared image, Faster R-CNN

## 0 引言

红外舰船目标检测在海上救援、海上军事活动等方面都具有十分重要的意义, 尤其在军事领域, 由于红外成像系统受光照条件影响小、成像稳定以及工作距离远等优点, 因此红外舰船目标检测对于海战场的战场监测与侦察具有深远的研究价值。目前基于深度学习的目标检测算法在近年来取得了很大的突破, 检测精度与速度与传统算法相比都有着质的飞跃, 并在各个方面都有着十分广泛的应用, 因此提出一种基于深度学习目标检测算法的红外舰船目标检测方法, 将深度学习的方法应用到红外成像领域。

目前主流的目标检测算法可以分成两大类<sup>[1]</sup>, One-stage 算法与 Two-stage 算法, R-CNN<sup>[2]</sup>是最早被提出的 Two-stage 算法, 同时也是将卷积神经网络应用到目标检测领域的开山之作, 但检测速度较慢, 随后提出的 Fast R-CNN<sup>[3]</sup>、Mask-R-CNN<sup>[4]</sup>与 Faster R-CNN<sup>[5]</sup>, 使目标检测的精度不断提高, Faster R-CNN 还创新性地提出并使用了 RPN (region proposal network) 网络以生成候选区域, Two-stage 类型的算法较之于 One-stage 算法速度相对较慢, 但是优点在于精度较高。One-stage 算法基于回归的思想, 使用 CNN 卷积特征, 直接产生物体的类别概率和位置坐标值 (无 region proposal), 经过单次检测可以得到

收稿日期: 2020-06-11; 修订日期: 2020-07-06.

作者简介: 顾佼佼 (1984-), 男, 博士, 讲师, 主要研究方向: 人工智能深度学习技术。

通信作者: 李炳臻 (1996-), 男, 硕士, 主要研究方向: 深度学习技术。E-mail: libingzhen123456@163.com.

最终的检测结果, 结构更加简洁, 比较典型的算法有 Redmon J.于 2015 年提出的 YOLO V1<sup>[6]</sup>、2016 年提出的 YOLO V2<sup>[7]</sup>、Liu W.提出的 SSD (single shot multi box detector) <sup>[8]</sup>, 2017 年提出的 DSSD (deconvolutional single shot multibox detector) <sup>[9]</sup>以及 2018 年提出的 YOLO V3<sup>[10]</sup>, 不足之处在于小目标检测精度不高并会出现漏检误检的问题。

深度学习目标检测算法发展迅速, 在红外图像领域也有所应用, 李慕锴在文献[11]中, 提出了一种基于 YOLO V3 的红外行人目标检测算法, 并在原有算法的基础上进行优化, 准确率达到了 85.89%; 崔少华在文献[12]中提出了一种改进的卷积神经网络用于检测红外行人目标, 选择 LeNet-7 作为检测网络, 提高了检测效率; 但是针对红外舰船目标的检测, 仍旧是基于传统方法, 向涛在文献[13]中提出了一种基于显著区域提取和目标精确分割相结合的红外舰船目标检测方法, 提升了检测的准确率; 邢莎在文献[14]中提出了基于局部边缘梯度特征分析的舰船目标检测方法, 通过采用自适应滤波、多尺度搜索等方法进行红外舰船目标检测; 宫剑在文献[15]中提出了一种基于引导滤波和自适应尺度局部对比度的舰船目标检测方法, 应用于红外偏振图像, 具有较高的准确性和鲁棒性; 上述红外舰船目标检测方法虽然都取得了不错的效果, 但是仍需手动提取特征, 耗时耗力, 因此本文提出了一种改进的 Faster R-CNN 目标检测算法, 对红外舰船目标进行检测, 并通过改进网络结构、基于数据集进行 Anchor 的改进以及优化损失函数的方法对原算法进行改进, 使得改进后的算法更适用于红外舰船目标检测。

### 1 Faster-R-CNN 算法原理

#### 1.1 设计思想

Faster R-CNN 是在 R-CNN 与 Fast R-CNN 的基础上进行改进得到的。在 Faster R-CNN 中, 引进了

Region Proposal Networks (RPN 网络) 代替 Selective search 生成候选区域检测框, 同时引入 Anchor box 的概念。简言之就是采用一个卷积神经网络来提取候选区域, 并将其与 Fast R-CNN 检测网络合并成一个网络进行训练和检测, 提高速度。

网络架构简图如图 1 所示。

具体的算法步骤为:

- 1) 使用一组基础卷积层提取待检测图片的 feature map (特征图), 该特征图同时被输入后续的网络与 RPN 网络, 共享卷积特征;
- 2) RPN 网络根据特征图生成 region proposals (候选区域), 输出候选框矩阵及其得分;
- 3) ROI pooling 层收集前步得到的 feature map 以及 RPN 传输的 region proposals, 池化操作得到 proposal feature map, 送入后续的全连接层;
- 4) 全连接层利用 proposal feature map 进行分类, 同时进行边框回归 (Bounding box regression) 得到最终检测框的位置以及得分信息。

Faster R-CNN 的主干网络是 VGG-16, 图 2 是 VGG-16 的参数列表。

VGG16 将卷积网络分成了 5 段, 每一段由两至三个卷积层和池化层组成, MaxPool 代表池化层, FC\_4096 代表全连接层, 4096 代表有 4096 个神经节点; Softmax 代表最后的分类器。一般都选取 VGG-16 作为 Faster R-CNN 的特征提取网络, 也可以采用 Res Net、Inception Net 等来代替。

#### 1.2 RPN 网络

Faster R-CNN 最大的创新点就是引入了 RPN 网络, 其主要作用是目标物体的精确定位, 利用 VGG16 的第五段卷积层 Conv5-3 上的特征图产生多个 Anchor 建议框, 再利用 Softmax 函数判断属于目标物体或者属于背景的概率, 引入 Anchor 的概念用以候选区域生成, Anchor 是 9 个由代码生成的矩形框, 涵盖了一

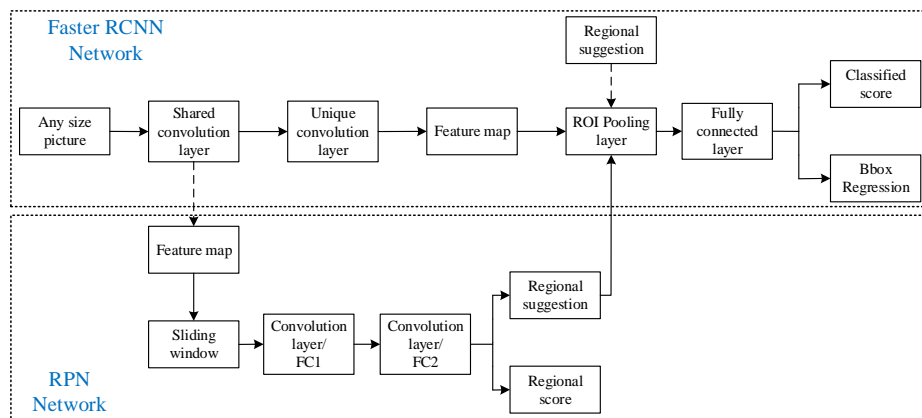


图 1 Faster R-CNN 网络结构图

Fig.1 Faster R-CNN network structure diagram

| Input Image (224,224,3) |         |        |              |
|-------------------------|---------|--------|--------------|
|                         | Size    | Number | Output       |
| Conv1-1                 | 3×3     | 64     |              |
| Conv1-2                 | 3×3     | 64     | (224,224,64) |
| MaxPool                 |         |        | (112,112,64) |
| Conv2-1                 | 3×3     | 128    |              |
| Conv2-2                 | 3×3     | 128    | (112,112,64) |
| MaxPool                 |         |        | (56,56,128)  |
| Conv3-1                 | 3×3     | 256    |              |
| Conv3-2                 | 3×3     | 256    | (56,56,128)  |
| Conv3-3                 | 3×3     | 256    | (56,56,128)  |
| MaxPool                 |         |        | (28,28,256)  |
| Conv4-1                 | 3×3     | 512    |              |
| Conv4-2                 | 3×3     | 512    | (28,28,256)  |
| Conv4-3                 | 3×3     | 512    | (28,28,256)  |
| MaxPool                 |         |        | (14,14,512)  |
| Conv5-1                 | 3×3     | 512    |              |
| Conv5-2                 | 3×3     | 512    | (14,14,512)  |
| Conv5-3                 | 3×3     | 512    | (14,14,512)  |
| MaxPool                 |         |        | (7,7,512)    |
|                         | FC_4096 |        |              |
|                         | FC_4096 |        |              |
|                         | FC_4096 |        |              |
|                         | Softmax |        |              |

图 2 VGG-16 网络参数列表图

Fig.2 VGG-16 network parameter list diagram

张图片上的所有区域。RPN 网络究其本质是通过卷积神经网络生成候选区域，输入任意尺寸的图像都可输出目标候选框矩阵及得分。具体示意图如图 3 所示：主干网络 VGG-16 对原输入图片进行卷积操作，提取特征并在第五段卷积之后得到该输入图片的卷积特征图，然后使用一个滑动窗口（即  $3 \times 3$  卷积核）在得到的特征图上进行滑动，得到一个 512 维的特征向量，并将此特征向量输入下一层的两个平行的全连接层进行分类和位置回归，由于每个滑动窗口的中心点对应  $k$  种 Anchor（在 Faster R-CNN 中， $k=9$ ），因此对于分类层来说输出  $2k$  个结果，即每个候选框的目标以及背景概率，而边框回归层共输出  $4k$  个结果，即每个候选框的位置坐标信息  $\{x,y,w,h\}$ ，并对得分信息进行非极大抑制（non maximum suppression, NMS）后输出得分较高的边框信息。

而 Anchor 则是一组固定大小的参考窗口，分为 3 组，面积分别为  $128^2$ 、 $256^2$ 、 $512^2$ ；而每种面积依据长宽比又分为 3 组，分别为 2:1、1:2、1:1，共有 9 个，相当于一组模板，在任意图像的任意位置都采用这组模板，如图 4 所示。

对每个滑窗位置可计算滑窗中心点对应原图中中心点，通过中心点和 Anchor 大小得到滑窗位置与原图位置的映射关系，继而让 RPN 网络学习此 Anchor 是否含有待检测物体。

### 1.3 边框回归 (Bounding box regression)

在 Faster R-CNN 中，采用边框回归来实现 Anchor box 到近似 Ground truth box (GT) 的过渡，示例图如

图 5 所示。图 5 中，虚线矩形框  $A$  代表原始的预测 Anchor，实线矩形框  $G$  代表目标的真实值 GT，而边框回归的作用就是将红色的框进行微调，并寻找一种函数关系，使得原始预测框经过映射得到一个跟真实值  $G$  更加接近的回归窗口  $G'$ ，也就是图 5 中的点划线预测框。

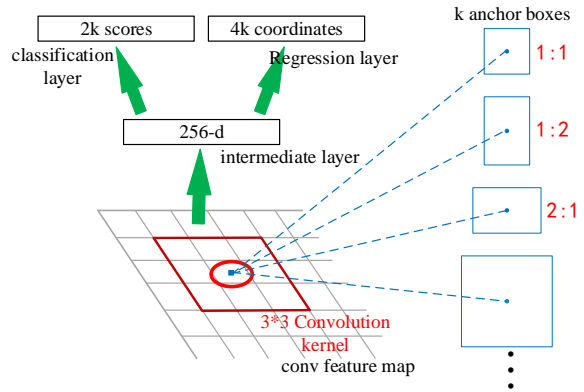


图 3 RPN 网络结构示意图

Fig.3 Schematic diagram of RPN network structure

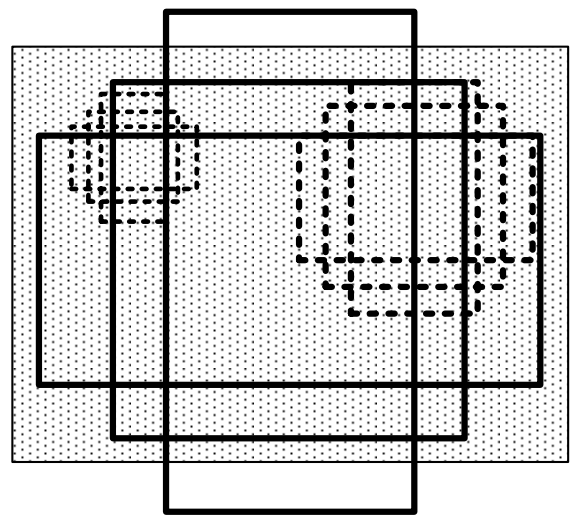


图 4 Anchor 示意图

Fig.4 Anchor schematic diagram

在图 5 中，给定  $A = (A_x, A_y, A_w, A_h)$ ， $GT = (G_x, G_y, G_w, G_h)$ ，Bounding box regression 则试图寻找一种映射关系  $f$ ，使得  $f(A_x, A_y, A_w, A_h) = (G'_x, G'_y, G'_w, G'_h)$ ，其中， $(G'_x, G'_y, G'_w, G'_h) \approx (G_x, G_y, G_w, G_h)$ 。通过采取平移和缩放变换得到最终的结果：首先对  $A$  做平移操作：

$$\begin{aligned} G'_x &= A_w dx(A) + A_x \\ G'_y &= A_h dy(A) + A_y \end{aligned} \quad (1)$$

接下来做缩放：

$$\begin{aligned} G'_w &= A_w \times \exp(dw(A)) \\ G'_h &= A_h \times \exp(dh(A)) \end{aligned} \quad (2)$$

上述公式中，需要学习得到的是  $dx(A)$ ， $dy(A)$ ，

$dw(A), dh(A)$ 这4个变换, 相应的平移因子 $(t_x, t_y), (t_x^*, t_y^*)$ 和缩放因子 $(t_w, t_h), (t_w^*, t_h^*)$ 如公式(3)所示:

$$\begin{aligned} t_x &= (x - x_a) / w_a, t_y = (y - y_a) / h_a \\ t_w &= \lg(w / w_a), t_h = \lg(h / h_a) \\ t_x^* &= (x^* - x_a) / w_a, t_y^* = (y^* - y_a) / h_a \\ t_w^* &= \lg(w^* / w_a), t_h^* = \lg(h^* / h_a) \end{aligned} \quad (3)$$

式中:  $(x, y, w, h)$ 、 $(x_a, y_a, w_a, h_a)$ 、 $(x^*, y^*, w^*, h^*)$ 分别对应预测框、Anchor box 以及 Ground truth box 中心点的坐标及宽和高。上述公式的意义就在于将 Anchor box 拟合回归到附近的真实边界框 GT, 并将经过回归变换后的 Anchor box 输出成为预测框。

## 2 改进的 Faster R-CNN 目标检测算法

### 2.1 改进的网络结构

在卷积神经网络中, 低层卷积主要提取目标的细节信息, 在网络位置中靠前, 高层卷积主要用来提取抽象的语义信息<sup>[16]</sup>, 在网络结构中位置比较靠后, 通过运行 CNN 解释器<sup>[17]</sup>得到卷积过程中的可视化特征图, 不同层级的特征图对比如图 6 所示。

其中, 图 6(a)为红外舰船原图, 图 6(b)为 VGG-16 中经过第 1 段卷积 Conv1 后输出的特征图, 图 6(c)为 VGG-16 中第 3 段卷积模块 Conv3 后输出的特征图, 图 6(d)为 VGG-16 中第 5 段卷积模块 Conv5 后输出的

特征图。从图 6 可以明显得看出, 低层卷积输出的特征图相比于高层卷积输出的特征图细节信息更加丰富, 特征也要更加明显。

基于此提出了一种多尺度提取特征的方法, 选取 VGG-16 作为主干网络, 去除全连接层, 利用 VGG-16 中的卷积层进行特征提取, 选取 3 段卷积后不同尺度的特征图进行特征拼接并输入 RPN 网络生成检测候选区域, 改进后的网络结构如图 7 所示。

在改进后的 Faster R-CNN 网络结构中, 选取 Conv1、Conv3、Conv5 卷积层后提取的特征图作为多尺度特征图, 以 Conv3 特征图为准, 并对 Conv1 的特征图进行池化操作缩小至相同尺寸、Conv5 的特征图进行反卷积增大至相同尺寸, 再进行特征拼接, 因为同一尺寸大小的特征图便于进行特征拼接, 然后输入至 RPN 网络进行候选区域生成, 拼接后特征图如图 8 所示。

从图 8 可以看出, 特征拼接使得待检测目标的细节信息变得更加丰富。并且对于 Conv5 特征图所进行的反卷积操作是卷积的反向过程, 也是一种上采样的方式, 但不同于基于插值的上采样方法, 反卷积操作可以在训练过程中反向传播学习、调整参数, 使得上采样参数更加合理。因此放大图片的尺寸的同时, 提供了一种最优的“上采样”方式, 使得反卷积过后的特征图较之前相比, 具有更加丰富的细节特征。

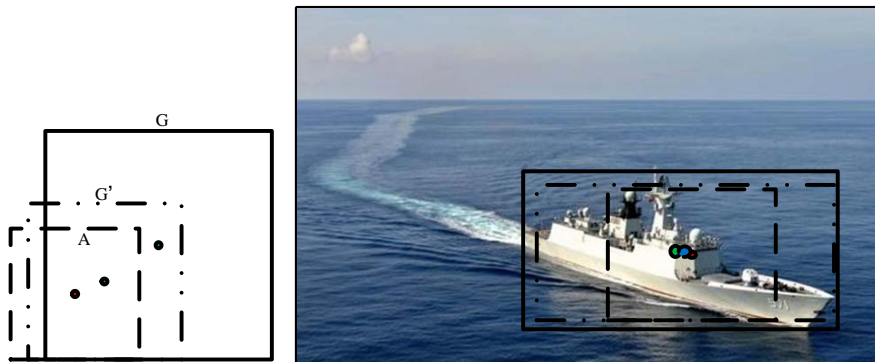


图 5 Bounding box regression 示例说明

Fig.5 Bounding box regression example description

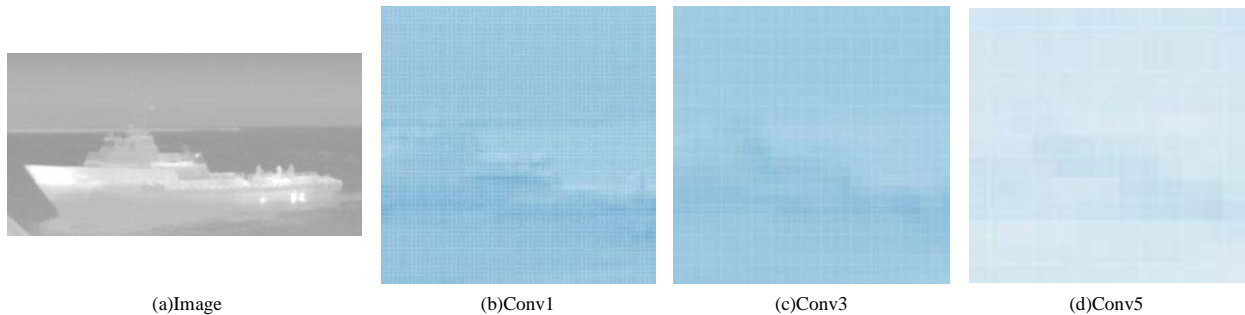


图 6 不同层级卷积后特征图对比

Fig.6 Comparison of characteristic graphs after convolution at different levels

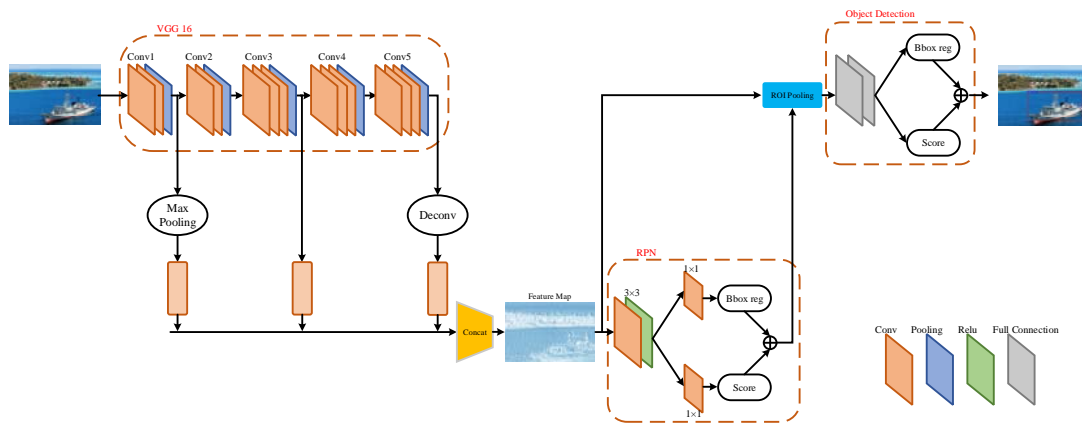


图 7 改进后网络结构图

Fig.7 Improved network structure diagram

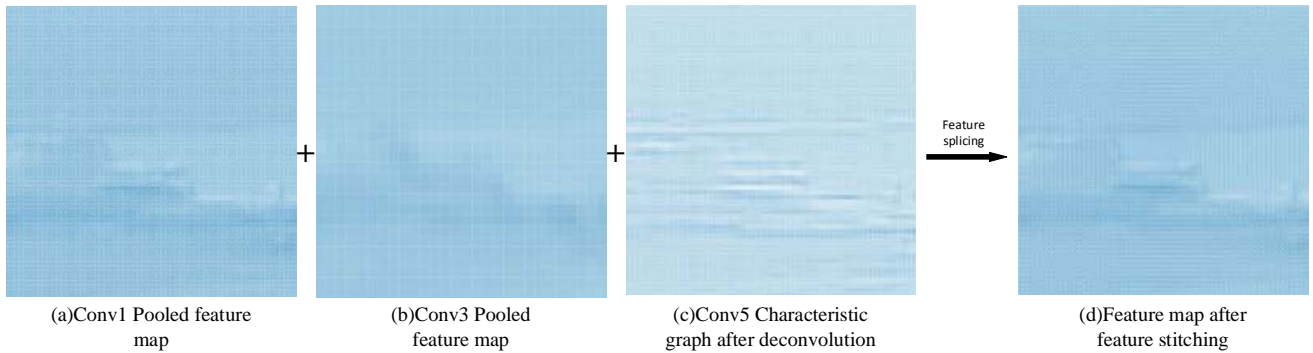


图 8 特征拼接后特征图

Fig.8 Feature map after feature stitching

在图 7 改进后的网络结构中，RPN 网络则根据特征拼接后的多尺度特征图生成 region proposals（候选区域），输出候选框矩阵及其得分，ROI pooling 层收集前步得到的多尺度特征图以及 RPN 传输的 region proposals，池化操作得到 proposal feature map，送入后续的全连接层，全连接层利用 proposal feature map 进行分类，同时进行边框回归（Bounding box regression）得到最终检测框的位置以及得分信息。

## 2.2 基于数据集的 Anchor 改进

Faster R-CNN 采用 Anchor box 作为每个滑窗位置的先验框并对目标物体进行检测，Anchor 的本质其实就是一组矩形框模板，本文在此基础上，结合红外舰船目标的特性，对 Anchor 的尺寸和个数进行改进，由于红外图片与可见光图片相比，具有分辨率低、对比度低、目标反射特征少等特点，目标与背景中一些细微的差异很难被分辨，因此原算法中，针对可见光图片中通用目标而设置的 9 种尺寸比例大小的 Anchor，并不适用于红外舰船目标的检测，对此本文针对红外舰船目标的特性，重新设置 Anchor 的尺寸和个数。

在原算法中，Anchor 的尺寸大小共有 9 种，考虑

到过多的 Anchor 对于目标检测会带来多余的计算，为避免冗余参数计算带来检测速度的变慢，将改进后的 Anchor 尺寸比例大小减少为两个尺度，长宽比为 3 种，共 6 种不同的 Anchor boxes:

首先增大 Anchor 初始感受野大小，用以弥补红外舰船目标反射特征少的特点，将 Anchor 初始感受野 base\_size 大小设置为  $32 \times 32$  (原算法中为  $16 \times 16$ )，并将 Anchor 感受野进行  $2^3=8$ 、 $2^4=16$  两种倍数的放大，参数设置为  $scales=2 * np.arange(3,5)$ ，此时  $16 \times 16$  区域变成  $256 \times 256$  与  $512 \times 512$ ，然后将宽高比例大小设置为 0.5、1、3，参数设置为  $ratios=[0.5,1,3]$ ，将上述的两个区域按 1:2、1:1、3:1 进行变换，共得到改进后的 6 个不同的 Anchor，如图 9 所示。

依据 Anchor box 和 Ground truthbox 的 IOU (Intersection over Union，意为交并比，即 Anchor box 与 GT (Ground truth box) 重叠面积与总面积的比例) 来进行下一步操作，如果  $IOU > 0.7$ ，则认为此 Anchor 中大概率包含目标，如果  $IOU \leq 0.7$ ，则认为此 Anchor 为背景，如果  $0.3 > IOU \geq 0.7$ ，则认为此 Anchor 没有作用，在检测及训练过程中丢弃，避免增加冗余计算参数。

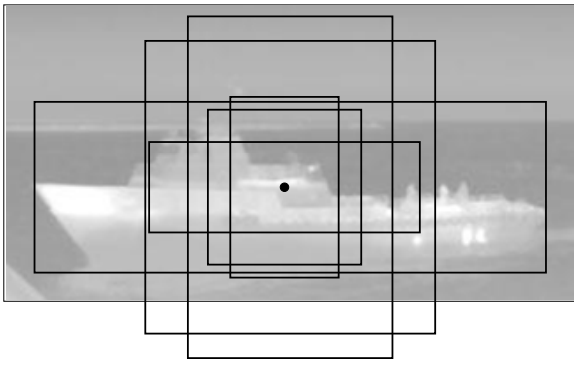


图 9 改进后的 Anchor 示意图

Fig.9 Improved Anchor schematic diagram

### 2.3 损失函数

整个损失函数由分类损失与回归损失两部分构成, 根据前文对网络结构的改进, 借鉴主干网络 VGG-16 与 R-CNN 的损失函数, 对 Faster R-CNN 的损失函数进行优化改进, 在回归损失中加入平方项, 使其在零点处导数变得更为平滑; 分类损失部分采用对数函数表达分类误差。改进后的损失函数具体如下所示:

1) 分类损失  $L_{cls}$

$$L_{cls} = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \quad (4)$$

$$= \frac{1}{N_{cls}} \sum_i -\lg [p_i^* p_i + (1 - p_i^*)(1 - p_i)]$$

在公式(4)中  $N_{cls}$  代表 Anchor 个数, 在网络训练过程中, 会选取 256 个 Anchor, 因此  $N_{cls}=256$ ;  $p_i$  代表第  $i$  个 Anchor 为预测目标的概率,  $p_i^*$  是 GT 的预测概率, 即当第  $i$  个 Anchors 与 GT 间  $IOU>0.7$ , 认为该 Anchor 中大概率包含目标,  $p_i^*=1$ ;  $L_{cls}(p_i, p_i^*)$  是目标与背景的损失, 选取对数函数表达这部分的误差。公式(4)的含义就是计算每一个 Anchor 的对数损失, 然后除以训练中挑选的总 Anchor 数, 得到平均损失。

2) 回归损失  $L_{reg}$

$$L_{reg} = \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (5)$$

在公式(5)中, 常见的回归误差一般选择平方损失函数, 但该损失函数误差惩罚过高, 因此在这里选择 smooth L1 loss 代表损失误差, 计算公式为:

$$L_{reg}(t_i, t_i^*) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L1}(t_i - t_i^*)$$

$$= \sum_{i \in \{x, y, w, h\}} \begin{cases} (\sigma x)^2 \times \frac{1}{2} & \text{if } |x| < \frac{1}{\sigma^2}, \\ |x| - 0.5/\sigma^2 & \text{otherwise} \end{cases}$$

在 RPN 网络中  $\sigma=3$ , 在 Fast R-CNN 中  $\sigma=1$ ; 而  $p_i^* L_{reg}(t_i, t_i^*)$  这一项代表着只有当  $p_i^*=1$ , 也就是有目标时, 才有回归损失, 其他情况下, 这一项均为 0;  $t_i =$

$(t_x, t_y, t_w, t_h)$  是一个向量, 代表预测框的 4 个坐标;  $t_i^*$  是预测框对应的真实值 GT 的 4 个坐标; 前面的参数  $\lambda$  则用来平衡回归损失与分类损失, 使两者差距不宜过大,  $N_{reg} \approx 2400$ ,  $N_{cls} \approx 256$ , 因此  $\lambda=10$ , 起到平衡权重的作用。

将上述分类损失与回归损失相加便可得整个网络的损失函数:

$$\text{Loss} = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

## 3 实验与分析

在这部分选用的训练集是从网上、工厂搜集的 1000 张包含舰船目标的红外图像, 测试集为 100 张包含大小各类红外舰船目标的图片, 分布在港口、码头、海洋等各个背景条件下。评价指标 mAP(mean average precision) 是对目标检测算法的检测精度进行效能评估的最常用的度量指标<sup>[18]</sup>, FPS (frame per second) 作为检测速度的评价指标, 选择这两项对训练好的模型进行精度和速度的评估。

### 3.1 模型训练

由于搜集到的数据集数量有限, 担心不足以支撑模型的训练, 因此对搜集到的数据进行数据增强处理, 通过左右镜像与上下镜像增强自己的训练数据集, 数据增强后示例如图 10 所示。

在图 10 中, 图 10(a)为红外舰船原图, 图 10(b)为原图经过左右镜像后得到的图片, 图 10(c)、图 10(d)为将上述两张图片进行上下镜像后得到的图片。通过运行 left\_right.py 与 up-down.py 对 1000 张数据集进行批量操作, 共得到数据增强后的图片集 4000 张。

实验环境的操作系统为 Windows10, 深度学习框架为 tensorflow1.9.0。实验硬件环境的 CPU 为 Intel(R) Core(R) i7 9700KF, 内存为 32 G。GPU 为 NVIDIA(R) GTX(R) 1080TI。训练过程中的参数设置: batch\_size=16, 预训练模型为 VGG-16, 初始学习率为 0.001, 衰减因子为 0.0005。首先将学习率设置成 0.001 训练 10000 次, 接下来在上一步训练的基础上将学习率设置成 0.0001 继续训练 10000 次, 再接着将学习率设置为 0.00001 继续训练, 使损失函数进一步收敛, 得到训练好的最终训练模型, 图 11 为训练过程中的损失函数图像。

从图 11 中可以看出在训练迭代 20000 次时, 损失函数已经收敛至 0.2 附近, 继续训练至 10000 次左右发现损失函数已稳定收敛于 0.2 上下, 由于改进后的网络结构增加了多尺度特征图从而引入了额外的参数, 担心继续训练下去会产生模型过拟合的问题, 于是在 30000 次停止训练, 得到训练结束的最终模型。

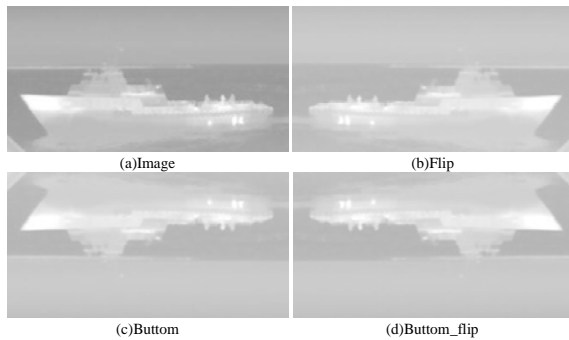


图10 数据增强示例图

Fig.10 Sample diagram of data enhancement

### 3.2 模型测试

通过对不同包含舰船目标红外图像进行测试，测试效果如图12所示，可以看到检测效果非常好，检测框上方是类别名称和置信度，训练好的模型可以准确识别出不同尺寸、不同大小、不同质量的图片的舰船目标。

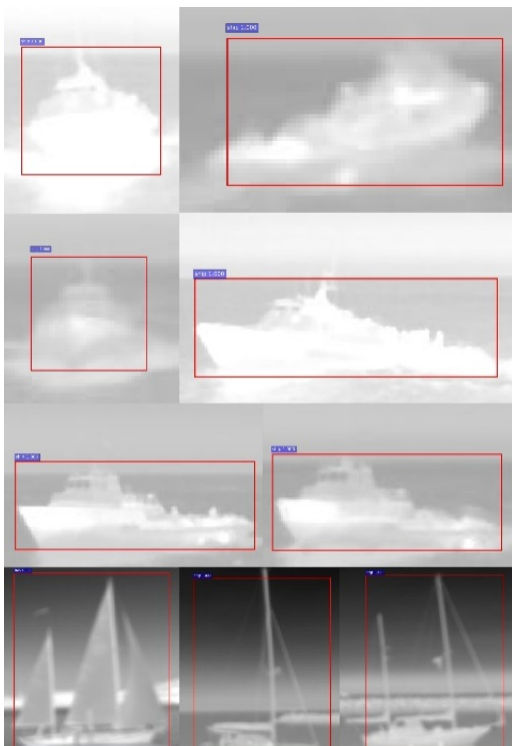


图12 红外舰船图像检测结果

Fig.12 Detection result of infrared ship image

### 3.3 结果分析

在目标检测算法中，评价一个算法模型的优劣最客观的评价指标有召回率(Recall)、准确率(Precision)、平均准确率AP、均值平均准确率mAP，通常使用均值平均精度(即mAP)来评估一个检测算法的检测准确度，数值越大检测效果越好，在本次实验中，目标类别只有一类，因此mAP值就等于AP值。

对于某一类问题，将分类的目标计为正例

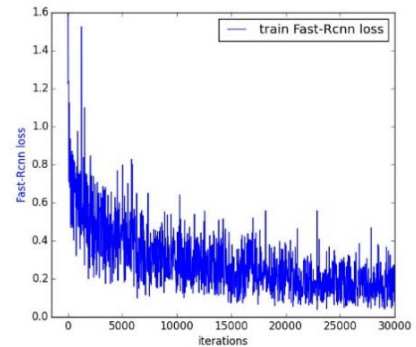


图11 改进的Faster R-CNN损失函数曲线

Fig.11 Improved Faster R-CNN loss function curve (positive) 和负例(negative)，那么就有如下的指标：

- 1) True positives (TP)：被正确地划分为正例的个数，即实际为正例而且被分类器判别出正例的数目；
- 2) False positives (FP)：被错误地划分为正例的个数，即实际为负例但被分类器判别为正例的数目；
- 3) False negatives (FN)：被错误地划分为负例的个数，即实际为正例但被分类器判别为负例的数目；
- 4) True negatives (TN)：被正确地划分为负例的个数，即实际为负例并且被分类器判别为负例的数目。

这些指标间的关系如表1所示。

表1 分类结果判别表

Table 1 Classification result discriminant table

| Real situation   | Discriminant result        |                           |
|------------------|----------------------------|---------------------------|
|                  | Positive example           | Counter example           |
| Positive example | TP(True positive example)  | FN(False Counter example) |
| Counter example  | FP(False positive example) | TN(True Counter example)  |

召回率R以及准确率P的定义公式分别为：

$$R = \frac{TP}{TP + FN}, P = \frac{TP}{TP + FP} \quad (6)$$

在公式(6)中，并在本次实验中，TP代表正确检测到舰的个数；FP代表误检测是舰船的个数；FN代表漏检测舰船的个数。AP值就为召回率与准确率所围成的R-P曲线面积，mAP值等于AP值；

使用原Faster R-CNN算法在同一数据集进行30000次的训练，训练结束后，在测试数据集上进行测试，对比改进前后检测效果；通过分别运行python工程文件mAP-master，对测试集进行计算得到改进前后的平均准确率AP值，并且得到R-P曲线，如图13所示。

图13中横坐标为召回率，纵坐标为相应的准确率，原Faster R-CNN算法在红外舰船测试集上的平均准确率AP=80.03%，经过改进之后在同一测试集上

的平均准确率  $AP=83.98\%$ ，在本次实验中  $mAP=AP$ 。改进前后的检测效果对比如图 14 所示。

31.28 s，改进后 FasterR-CNN 检测 100 张图片所花费时间为 33.84 s，其具体的性能对比如表 2 所示。

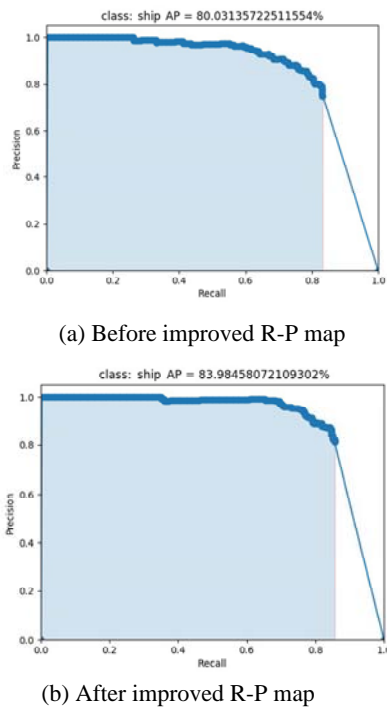


图 13 改进前后 Faster R-CNN 在红外舰船测试集上的 R-P 曲线

Fig.13 R-P curves of Faster R-CNN on infrared ship test set before and after improvement

筛选出的这两组图像均可以很好地表现改进算法的相比于原有算法的改进成效，从图 14 前 4 组对比图片中可以明显看出，改进后的算法在在选红外目标的位置准确性方面有了很大的提升，这是因为基于红外数据集改进的 Anchor 尺寸更适合红外舰船目标，从而使得最后输出的预测框能够更为准确地框选红外目标；同时在检测红外目标精度方面也有了很大提升，如图 14 最后两组对比图片所示，这是由于改进后网络结构中增加的多尺度特征图使得模型可以提取到更为丰富的红外目标特征信息；原算法中容易出现重复检测的问题也得到了改善，如图 14 第 5、6 组对比图片所示，一方面，这是由于网络结构中增加了多尺度特征图从而提高了特征提取能力，另一方面，对训练数据集所做的数据增强处理在某种程度上也使得重复检测的问题得到了缓解；综上所述，改进后的算法在框选目标位置准确性、检测准确率上有了很大程度的提升，同时也改善了原算法中容易出现重复检测问题。

对改进前后的两种算法，运行 python 工程文件 mAP-master 之后可得到检测 100 张图片所花费的时间，原 Faster R-CNN 检测 100 张图片所花费时间为



图 14 Faster R-CNN 改进前后红外目标检测效果对比  
Fig.14 Comparison of infrared target detection effect before Faster R-CNN improvement

从表中可以看出，原 Faster R-CNN 算法的  $mAP$  值为 80.03%，而改进后算法的  $mAP$  值为 83.98%，较



之于改进前提升了 3.95%，精度提高的同时，检测每张图片所花费的时间较之于改进前稍有变长，这是因为在网络结构中增加了 3 个特征图进行拼接形成多尺度特征图用于加强对红外目标的特征提取从而引入了额外的计算参数所导致的。

表 2 改进前后算法性能对比

Table 2 Comparison of algorithm performance before and after improvement

| Model name            | AP/%  | mAP/% | Time/s |
|-----------------------|-------|-------|--------|
| Faster R-CNN          | 80.03 | 80.03 | 0.3128 |
| Improved Faster R-CNN | 83.98 | 83.98 | 0.3384 |

#### 4 结论

提出了一种改进 Faster R-CNN 的红外舰船目标检测算法，对网络结构和 Anchor 进行改进，同时优化了损失函数。该算法应用于红外舰船目标检测时，检测精度可以达到 83.98%，较原算法相比提高了 3.95%；对比算法改进前后的红外图像检测结果，验证了改进方法的可行性。本文研究的改进方法使得红外舰船目标的检测效果大大提升，可将改进后的算法模型应用在军事及工业领域，工业领域方面可应用于夜晚或严重光照不足条件下的码头、港口过往船舶检测，对海上救援、海上交通都有着重要的应用研究价值；在军事领域方面不仅可以应用于海上战场进行舰船目标检测，还可装备应用于红外反舰武器系统上，提升打击精度，从而提升部队战斗力。

#### 参考文献:

[1] 施泽浩, 赵启军. 基于全卷积网络的目标检测算法[J]. 计算机技术与发展, 2018(5):55-58.  
SHI Zehao, ZHAO Qijun. Target detection algorithm based on full convolution network[J]. *Computer Technology and Development*, 2018(5): 55-58.

[2] Uijlings J R R, Sande K E A V D, Gevers T, et al. Selective search for object recognition[J]. *International Journal of Computer Vision*, 2013, **104**(2): 154-171.

[3] Girshick R. Fast R-CNN[C]//*Proceedings of ICCV*, 2015, **2015**: 1440-1448.

[4] Kaiming H, Georgia G, Piotr D, et al. Mask R-CNN[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, **18**(11): 1-1.

[5] Ren S, He K, Girshick R, et al. Faster r-cnn: towards real-time object detection with region proposal networks[C]//*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, **39**(6): 1137-1149.

[6] Redmon J, Divvala S, Girshick R, et al. You only look once: unified,

real-time object detection[C]//*Proceedings of CVPR*, 2015: 779-788.

[7] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger[C]//*IEEE Conference on Computer Vision and Pattern Recognition*, 2017, **12**(7): 6517-6525.

[8] LIU W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector[C]//*Computer Vision-ECCV 2016*, Cham: Springer, 2016, **9905**: 21-37.

[9] FU C Y, LIU W, Ranga A, et al. DSSD: deconvolutional single shot detector[C]//*2017 IEEE International Conference on Computer Vision*, 2017: 2999-3007.

[10] Redmon J, Farhadi A. YOLO v3: An incremental improvement [C]//*IEEE Conference on Computer Vision and Pattern Recognition*, 2018, **24**(9): 2458-2476.

[11] 李慕轲, 张涛, 崔文楠. 基于 YOLOv3 的红外行人小目标检测技术研究[J]. 红外技术, 2020, **42**(2): 176-181.  
LI MUYI, ZHANG TAO, CUI WENNAN. Research on infrared pedestrian small target detection technology based on YOLO v3[J]. *Infrared Technology*, 2020, **42**(2): 176-181.

[12] 崔少华, 李素文, 黄金乐, 等. 改进的 CNN 用于单帧红外图像行人检测的方法[J]. 红外技术, 2020, **42**(3): 238-244.  
CUI SHAOHUA, LI SUWEN, HUANG JINLE, et al. Improved CNN square method for human detection of single-frame infrared image[J]. *Infrared Technology*, 2020, **42**(3): 238-244.

[13] 向涛. 一种基于显著区域提取的红外图像舰船目标检测方法[J]. 电讯技术, 2020, **60**(7): 50-56.  
XIANG TAO. A ship target detection method based on salient region extraction in infrared image[J]. *Telecommunication Technology*, 2020, **60**(7): 50-56.

[14] 邢莎, 吉林, 雍杨, 等. 基于梯度统计特性的自动红外舰船目标检测[J]. 数字技术与应用, 2013(10): 66-68, 70.  
XING SHA, JI LIN, YONG YANG, et al. Automatic infrared ship target detection based on gradient statistics[J]. *Digital Technology and Applications*, 2013(10): 66-68, 70.

[15] 宫剑, 吕俊伟, 刘亮, 等. 红外偏振图像的舰船目标检测[J]. 光谱学与光谱分析, 2020, **40**(2): 586-594.  
GONG JIAN, LV JUNWEI, LIU LIANG, et al. Ship target detection based on infrared polarization image[J]. *Spectroscopy and Spectral Analysis*, 2020, **40**(2): 586-594.

[16] 吴天舒, 张志佳, 刘云鹏. 基于改进 SSD 的轻量化小目标检测算法[J]. 红外与激光工程, 2018(7): 37-43.  
WU TIANSHU, ZHANG ZHIJIA, LIU YUNPENG. Lightweight small beacon detection algorithm based on improved SSD[J]. *Infrared and Laser Engineering*, 2018(7): 37-43.

[17] WANG Zijie J, Robert Turko, Omar Shaikh, et al. CNN explainer: learning convolutional neural networks with interactive visualization[J]. *Journal of Sensors*, 2019(6): 1-13.

[18] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]// *Proceedings of the European Conference on Computer Vision*, Berlin, Heidelberg: Springer, 2014, **8689**: 818-833.