Modeling and Simulation 建模与仿真, 2024, 13(1), 50-60
Published Online January 2024 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2024.131006

Hans汉斯

基于深度学习的视频船舶目标追踪模型

卞昌硕[1]，余胜任[1]，阳书威[2]，段玮鹏[3]

[1]南京信息工程大学数学与统计学院，江苏 南京
[2]武汉工程大学计算机学院，湖北 武汉
[3]中北大学数学学院，山西 太原

收稿日期：2023年11月2日；录用日期：2023年12月31日；发布日期：2024年1月9日

摘　要

船舶追踪是保证水上船舶交通安全的重要技术之一，船舶目标追踪模型可以对船舶的轨迹建模从而预测船只位置，监控船只交通安全，然而在船舶在交汇情境下，由于船舶被部分遮挡或处于复杂背景中，仅依赖视觉特征进行船舶追踪可能会导致追踪边界的不准确定位等问题。此外，当船只完全被遮挡并再次出现时，传统ReID方法往往不能为这些船只正确分配轨迹。为了解决这些问题，本文首先提出了一种融合光流场特征的单目标船舶追踪模型。该模型通过将视觉追踪特征与光流特征相结合，实现了更精确的目标边界定位。接着，通过结合Yolov5目标检测模型和经过改进的单目标追踪模型，实现了一种判别式多目标追踪模型，并采用更有效的匹配方法，一定程度上缓解了目标消失再出现id丢失的问题。实验表明，融合光流网络的目标追踪模型能更好地定位目标，而本文的联合多目标追踪框架在目标定位和ID分配方面均优于DeepSort、ByteTrack等追踪模型。

关键词

深度学习，船舶追踪，光流

Video Ship Target Tracking Model Based on Deep Learning

Changshuo Bian[1], Shengren Yu[1], Shuwei Yang[2], Weipeng Duan[3]

[1]School of Mathematics and Statistics, Nanjing University of Information Science & Technology, Nanjing Jiangsu
[2]College of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan Hubei
[3]School of Mathematics, North University of China, Taiyuan Shanxi

Received: Nov. 2nd, 2023; accepted: Dec. 31st, 2023; published: Jan. 9th, 2024

Abstract

Ship tracking is one of the crucial technologies for ensuring the safety of maritime traffic. Ship tar-

文章引用: 卞昌硕, 余胜任, 阳书威, 段玮鹏. 基于深度学习的视频船舶目标追踪模型[J]. 建模与仿真, 2024, 13(1): 50-60. DOI: 10.12677/mos.2024.131006

get tracking models can model the trajectory of ships to predict their positions and monitor maritime traffic safety. However, in situations where ships intersect, issues such as partial ship obstructions or complex backgrounds can lead to inaccurate tracking boundary positioning when relying solely on visual features. Additionally, traditional discriminative tracking models often struggle to correctly assign identities to ships when they reappear after being completely obscured. To address these challenges, this paper first introduces a single-object ship tracking model that integrates optical flow field features. This model combines visual tracking features with optical flow features, resulting in more precise target boundary localization. Subsequently, a joint multi-object tracking framework is implemented by combining the Yolov5 object detection model with an improved single-object tracking model. This framework also utilizes more effective matching methods, partially alleviating the issue of losing IDs when targets reappear. Experimental results demonstrate that the target tracking model incorporating optical flow performs better in target localization. Furthermore, the joint multi-object tracking framework presented in this paper outperforms tracking models like DeepSort and ByteTrack in both target localization and ID assignment.

Keywords

Deep Learning, Ship Tracking, Optical Flow

1. 引言

水上交通是我国的重要交通方式，我国有着极其发达的水上交通运输业，船舶追踪在水运交通领域具有至关重要的意义。它被广泛应用于港口和其他重要水运交通场景。近年来，计算机视觉领域的目标追踪技术广泛应用于各行业。特别是在船舶监控领域，目标追踪技术的应用具有重要意义。

船舶在交汇情况下存在着严重遮挡或复杂背景等问题，这对多目标追踪时的目标定位精准度以及 id 分配产生了巨大的挑战，仅仅依赖视觉特征很难对目标准确定位，另外，船舶消失再出现时往往被严重遮挡，使用传统的 ReID 方法很难对新目标分配正确的 id。

传统的目标追踪模型分为生成式与判别式模型，生成式模型为当前目标区域进行建模，寻找图像内与目标相似度最高的区域，判别式追踪模型的主要特点是它关注于直接学习目标对象和背景之间的差异，以便有效地定位和跟踪目标。传统的追踪模型包括光流法、Kalman 滤波法和 Meanshift 方法。光流法通过分析像素之间的位移来估计目标的运动，Kalman 滤波法利用动态模型和测量数据对目标运动轨迹建模从而预测下一帧目标位置，而 Meanshift 方法则通过颜色直方图的变化来跟踪目标的位置和大小。这种传统的追踪模型并不能应对船舶跟踪中复杂的场景。但是这些追踪模型的思路依然可以为实际工程应用时作为深度学习模型的有益的辅助手段。

近年来，由于深度学习的快速发展，基于深度学习的视觉目标追踪模型成为了目标追踪的主流方法，例如，Bewley 等人提出的 SORT [1]，与 DeepSORT [2]方法，这类模型通过利用卡尔曼滤波对目标的运动轨迹建立运动模型，通过联级匹配为检测到的目标确认轨迹，zhang 等人提出了 Bytetracker [3]，利用低置信度的预测框对消失目标进行二次匹配。这些方法利用滤波器来预测目标的下一帧位置并维护目标信息。通过独特的匹配机制，将当前帧的目标检测结果与之前帧的预测结果进行合理匹配。这是多目标船舶追踪的一般范式，最近两年，各种创新的多目标模型也被陆续提出。Sun 等人[4]提出了 TransTrack，

这是一种联合检测与追踪的 JDE 范式，使用查询－键机制进行多目标追踪，Meinhardt 等人[5]提出了 Trackformer，以一种新的 tracking-by-attention 范式实现了一种无缝的帧间数据关联，注意力机制确保了模型同时考虑位置、遮挡和目标的识别特征。Vaquero 等人[6]提出了 SiamMOTION，通过注意力机制与特征金字塔网络提供感搜索区域高质量的特征。之后，搜索区域特征与目标模板特征被送入一个成对的深度区域建议网络和多对象惩罚模块用来生成高质量的多目标预测。Cai 等人提出了 MeMOT [7]模型，它是在一个公共框架下执行对象检测和数据关联，能够在长时间跨度后链接对象。这是通过保留一个大的时空内存来存储被跟踪对象的 ID Embeddings，并根据需要自适应地从内存中引用和聚合有用的信息来实现关联。

相比之下，单目标追踪模型对第一帧目标位置进行手动标注，之后追踪模型通过第一帧的视觉信息在随后的视频帧中寻找目标位置，典型的方法包括 SiamFC [8]，DiMP [9]等，这类方法通过对搜索区域与目标区域提取特征，使用卷积操作进行相似度计算寻找目标位置。但是，基于卷积操作的相似度计算方法难以使用全局信息寻找目标位置，因此 Stark [10]把一个 Transformer [11]网络引入代替卷积操作计算相似度。进而 SimTrack [12]在 stark 的基础上使用 VIT [13]特征提取器提取特征。然而，在涉及船舶被遮挡或船舶交汇等复杂场景中，仅基于外观特征的检测方法难以正确标定目标的边界，从而降低了追踪的准确性。

本研究提出了一种融合光流特征和视觉追踪特征的目标追踪模型，通过对相邻帧间的光流信息建模来指导 Stark 追踪模型，利用融合了运动信息和外观信息的特征来预测目标的位置，有效缓解了边界定位问题。此外，通过联合 YOLO 目标检测模型和 Stark 单目标追踪模型，实现了一种联合的多目标追踪方法。

2. Yolov5-OF-Stark 多目标追踪模型

本研究通过将 Stark 单目标追踪器与光流计算网络相耦合，同时计算目标的视觉追踪信息和运动信息，并将两种信息融合以预测追踪目标的坐标，以获得更为鲁棒的目标位置信息。进而，通过联合使用 Yolov5 目标检测器和改进的 Stark 目标追踪器，成功完成了多目标追踪任务。

2.1. 融合光流特征的 OF-Stark 目标追踪模型

2.1.1. Stark 模型

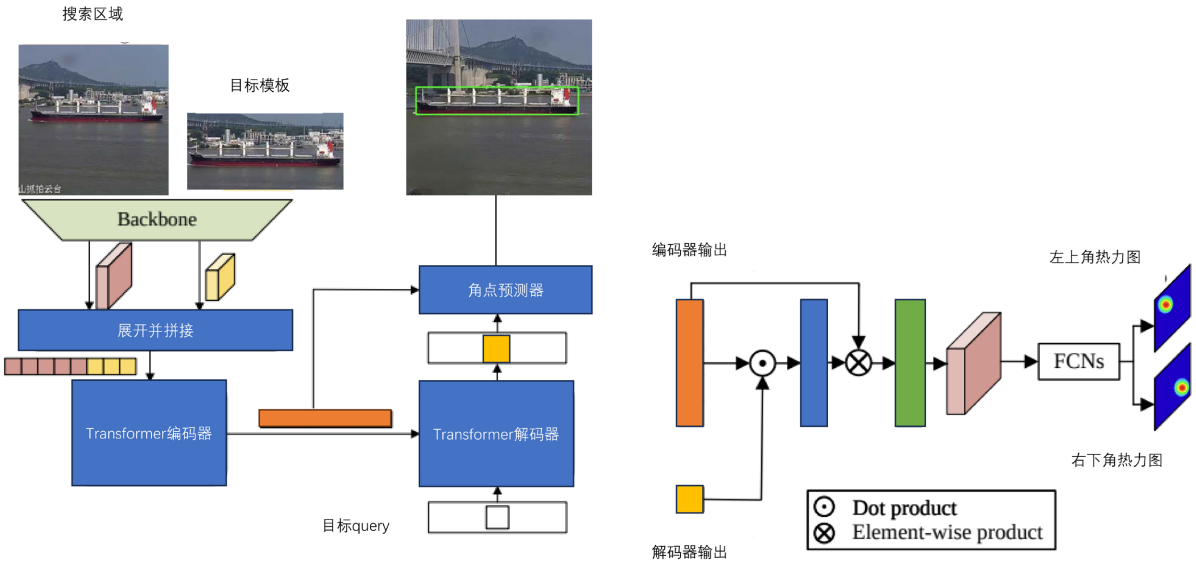


Figure 1. Stark network architecture diagram
图 1. stark 模型结构图

Stark 是由 Yan 等人于 2021 年提出的目标追踪器。基于孪生网络的目标追踪器通过目标的外观特征在视频的所有帧中搜索目标的位置，这一方法将检测与追踪相融合在一起。传统的孪生网络结构使用卷积核在搜索区域上搜索目标，然而这种方式仅建立了目标与搜索区域的局部联系，无法全面捕捉目标的全局信息。相比之下，Stark 在搜索步骤中引入了 Transformer [11]结构来代替卷积核执行相似度匹配操作，从而实现了全局模板与搜索区域的信息关联。其结构图如图 1 所示。

在本文采用 Stark-s 模型作为基线，Stark 模型接受目标模板与搜索区域作为输入，在进入孪生网络提取特征后得到的特征图被展开拼接后送入 transformer 编码器。而 transformer 解码器接受目标查询与编码器的输入为输出。此外，stark 还设计了一种新的输出头接受 transformer 解码器与编码器的输出，这是一种注意力机制，在 transformer 编码器的输出与解码器的输出计算相似度后将搜索区域与相似度特征元素相乘以增强重要区域。之后得到的追踪特征被送入角点预测器预测目标的左上角与右下角。

2.1.2. OF-Stark 模型

在目标追踪中，基于孪生网络的目标追踪器通常依赖目标的外观特征来定位物体，然而，在船只交汇等场景中，由于目标船只可能被遮挡或背景变得复杂，这种方法往往难以准确确定目标的边界。此外，在目标追踪过程中，追踪器通常在视频的当前帧搜索区域上进行相似度匹配，而这种单帧匹配的方式忽略了视频中时间维度所蕴含的信息。

考虑到船舶追踪场景下的特殊性，光照和摄像头位置角度在短时间内很少发生变化。因此，本文采用了一种创新的方法，对两帧间的光流信息建模，以获取相邻两帧之间目标的运动信息。通过融合外观特征与光流特征，本研究实现了更准确的目标定位。改进后的 Stark 模型结构如图 2 所示。

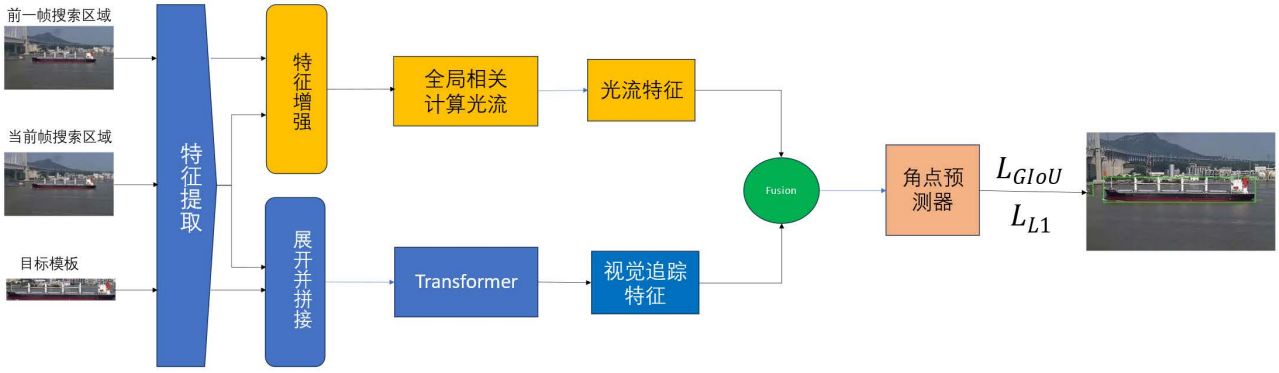


Figure 2. OF-stark network architecture diagram
图 2. OF-stark 模型结构图

首先，追踪模型接受目标的初始坐标作为输入，这些坐标用来标记目标的初始位置，并依赖这些位置，模型为每个目标初始化模板特征。以一个目标为例，在初始化阶段，模型借助 Yolov5 检测到的坐标框来确定目标的初始位置。由于光流的引入，需要确保在整个视频序列中，两个搜索区域的相对位置保持不变，因此前一帧的搜索区域也由前一帧的目标坐标决定。

随后，目标模板首先被送入 ResNet 进行特征提取，然后在每一帧进行追踪时，模型接受前一帧和当前帧的搜索区域作为输入。在追踪过程中，首先将当前帧的搜索区域特征与模板特征拼接，再加上位置编码，然后将其输入到由 Stark 设计的 Transformer 模型中执行相似度匹配，以获取深度特征用于视觉追踪：

$$L = \text{Transformer}\left(\left(F_t + P_t\right),\left(F_n + P_n\right)\right) \in \mathbb{R}^{H\times W\times c} \tag{1}$$

其中，F_t，F_n 是目标特征与当前帧搜索区域特征，P_t，P_n 是位置编码。L 是视觉追踪的深度特征。长宽

和通道数分别为 H,W,c。

受到 Xu 等人的启发[14]，本文采用通过计算像素与其全局关系的相似度的方式，对视频中的光流信息建模，具体地说，图 2 上的黄色部分是用来计算光流场的模型，依照 gmflow 中的方法，首先，两个相邻帧的搜索区域被送入 resnet 模型用来提取深度特征，在此之后，这两块特征图被送到特征增强模块，这是一个堆叠的六个自注意力和交叉注意力模型，用来关联这两个相邻帧的特征。最终获得的相邻两帧特征为 F_n， F_{n-1}，长宽和维度分别为 H,W,D。光流是相邻两帧每个像素的运动趋势，通过计算相邻两帧特征图的像素的相似度之差可以估计两帧的光流信息。这两个特征被送入图 2 中的全局相关模块来计算光流，首先通过计算相关性比较 F_1 中每个像素对于 F_2 中所有的像素特征相似性：

$$C=\left(F_nF_{n-1}^{\mathrm{T}}\right)\Big/\sqrt{D}\in\mathbb{R}^{H\times W\times H\times W} \tag{2}$$

通过计算 F_n， F_{n-1} 中每个像素特征的点积再除以 $\sqrt{D}$ 进行归一化，得到了两帧图像深度特征的相关矩阵 C。之后，使用 softmax 运算对 C 的最后两个维度进行归一化：

$$M=\mathrm{softmax}\left(c\right)\in\mathbb{R}^{H\times W\times H\times W} \tag{3}$$

M 给出了 F_n 中每个位置相对于 F_{n-1} 中所有位置的匹配分布，然后，通过匹配分布 M 对像素网格 $G\in\mathbb{R}^{H\times W\times 2}$ 的 2D 坐标进行加权，可以获得对应的坐标关系 $\hat{G}$。

$$\hat{G}=MG\in\mathbb{R}^{H\times W\times 2} \tag{4}$$

最后，光流 V 通过计算相应像素坐标之差获得：

$$V=\hat{G}-G\in R^{H\times W\times 2} \tag{5}$$

2.1.3. 特征融合

得到了搜索区域的光流特征后，希望视觉追踪特征在送入角点预测器之前能够注意到运动趋势较大的区域，并且学会利用这种运动趋势信息得到更精确的目标定位。故本文设计了一个特征融合模块用来融合光流特征与追踪的特征，具体的说，Fusion 模块是一个注意力模块，它实现了将视觉追踪特征与光流特征结合的操作。具体结构如图 3 所示：

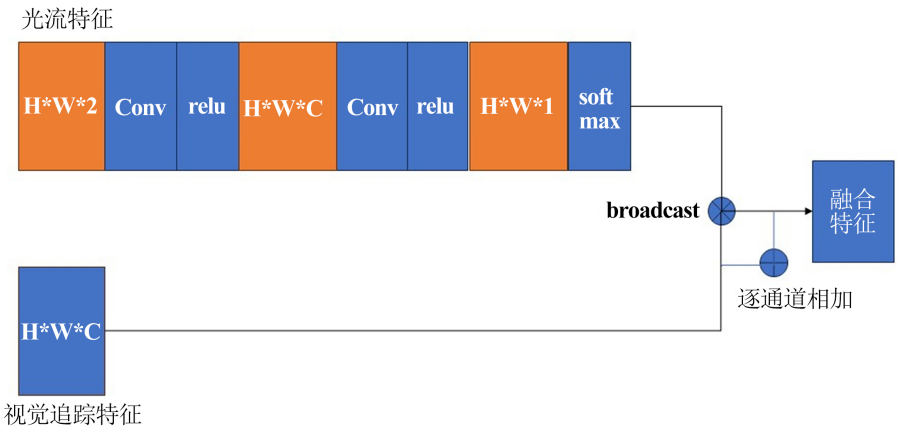


Figure 3. Fusion module structure diagram
图 3. Fusion 模块结构图

首先，光流特征是一个两维的特征，代表了每个像素点在横向与纵向的运动趋势，在与视觉追踪特征融合之前，它首先被一个 1*1*c 的卷积块映射到高维空间，之后被另一个 1*1 的卷积核压缩到一维空间，这使得运动特征可以通过训练来更好地与视觉追踪特征融合：

$$\hat{V}=\mathrm{relu}\left(\mathrm{conv}\left(\mathrm{relu}\left(\mathrm{conv}\left(V\right)\right)\right)\right)\in\mathbb{R}^{H\times W} \tag{6}$$

之后执行 Softmax 归一化操作，本文希望运动特征能够指导视觉追踪特征注意到运动趋势较大的区域，于是接下来我执行 broadcost 操作，把视觉追踪特征的每一个通道与归一化的运动特征对应元素相乘，所得到的结果与原始的视觉追踪特征对应通道相加，

$$\hat{L}=L+L\otimes\mathrm{softmax}\left(\hat{V}\right) \tag{7}$$

最终的融合特征 $\hat{L}$ 被送入角点预测器来得到目标最终的定位。

2.1.4. OF-Stark 训练策略

OF-stark 追踪器是以端到端的方式进行训练的。遵循 Stark 的训练损失，本文使用 L1 损失与 GIOU 损失的联合损失训练 OF-stark 追踪网络：

$$L=\lambda_{\mathrm{GIoU}}L_{\mathrm{GIoU}}\left(b_i,\hat{b}_i\right)+\lambda_{\mathrm{L1}}L_{\mathrm{L1}}\left(b_i,\hat{b}_i\right) \tag{8}$$

其中 λ_{GIoU}， λ_{L1} 是超参数， $b_i,\hat{b}$ 分别代表坐标的真值与预测的坐标。

GIOU 损失的表达式如下：

$$L_{\mathrm{GIoU}}=1-\mathrm{IoU}+\frac{\left|C\setminus\left(b_i\cup\hat{b}_i\right)\right|}{|C|} \tag{9}$$

IOU 是真实框与预测框的交并比，C 是真实框与预测框的最小闭包框。

L1 损失的表达式如下：

$$L_{\mathrm{L1}}=\frac{\sum\left|b_i-\hat{b}_i\right|}{n} \tag{10}$$

L1 损失对于每个预测框的四个坐标值以及每个真实目标框的相应坐标值求平均差值来衡量真实值与预测值的位置差。

在训练过程中，每一个元组以模板，当前帧搜索区域，前一帧搜索区域送入模型，当前帧搜索区域与前一帧搜索区域统一由当前帧目标坐标框定，保证当前帧搜索区域与前一帧搜索区域在原图上的搜索区域相对位置不变性。并做相同的预处理。在追踪过程中，模板坐标由 Yolov5 检测的坐标提供，并随着 Yolov5 的检测到此目标更新的状态，目标模板会保持更新以适应外观变化。

2.2. 联合多目标追踪框架

2.2.1. 目标检测

单目标追踪器在进行追踪时通常需要手动指定第一帧中的目标作为模板，然后在后续帧中计算相似度以寻找并跟踪目标。这种单目标追踪器并不能直接应用到多目标追踪，因此，本文使用目标检测器提供目标位置的先验信息。

目标检测旨在图像或视频中检测和定位物体的位置，并为这些物体分配相应的类别标签。与图像分类不同，目标检测不仅需要识别物体的类别，还需要确定其在图像中的位置，通常以矩形边界框的形式表示。目标检测方法通常可以分为一阶段目标检测器和两阶段目标检测器两类。

一阶段目标检测器，如 YOLO [15]、DETR [16]，SSD [17]，RetinaNet [18]，一次性完成目标检测的所有步骤，包括物体位置的检测和类别的分类，而无需额外的候选区域生成步骤。另一方面，二阶段目标检测器，如 R-CNN [19]、Fast R-CNN [20]等，将目标检测任务分为两个独立的阶段：候选区域生成和

物体分类与定位。这两个阶段分别负责生成可能包含目标的候选区域，并对这些候选区域进行物体检测和分类。

考虑到运行效率和检测精度的平衡，本文选择了 Yolov5 作为目标检测器，以提供追踪模型目标位置的先验信息。在进行多目标追踪任务时，首先使用 Yolov5 来定位视频中各艘船只的初始位置，并为每个检测到的目标分配唯一的 ID 以标识其身份信息。随后，这些目标被送入 OF-stark 用来预测每个目标在接下来几帧的位置。在追踪模型运行一定数量的帧之后，再次使用 Yolov5 检测船只的位置，然后将追踪结果与检测结果进行匹配，以确定目标是否消失或新目标是否出现，并更新目标的模板特征，以适应目标在追踪过程中产生的外观变化。

然而，目标检测器通常无法区分不同目标之间的关系，因此在进行检测时，我需要将追踪结果与检测结果进行匹配。为了完成这一过程，本文设计了一种简单的匹配策略。

2.2.2. 匹配策略

为了使追踪模型能够适应新目标出现，目标消失与目标重现等情况，本文设计了两步目标匹配策略，如图 4 所示。其中，追踪结果中 1 号为消失的目标。3 号为即将出现的新目标，在新检测结果中分别对应着蓝色与橙色坐标框对应的目标。

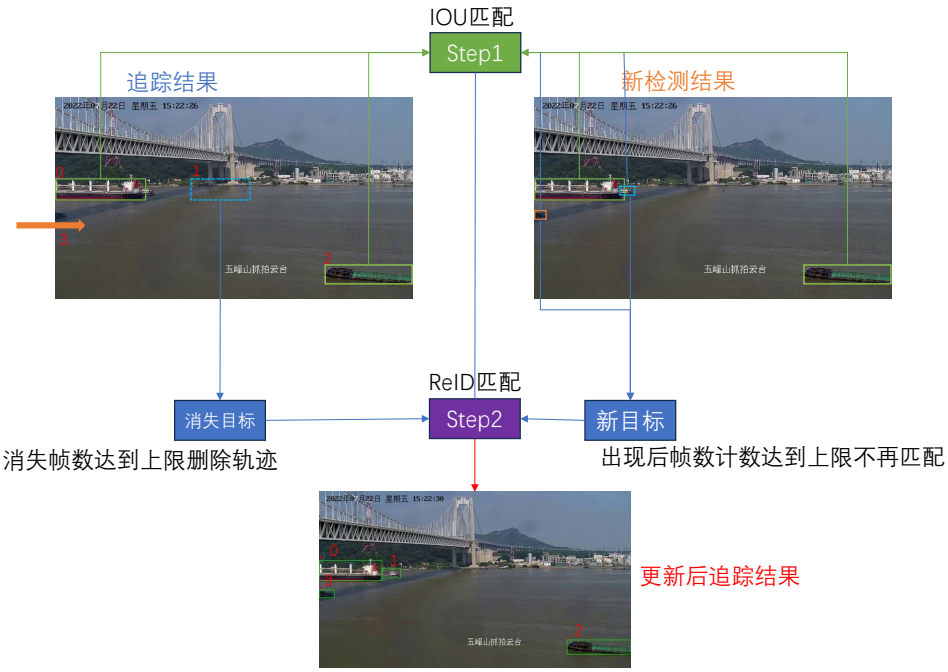


Figure 4. Matching strategy
图 4. 匹配策略

Step1：IOU 匹配

目标对于追踪结果和检测出现的目标，首先采用 IOU 匹配的方式进行匹配，计算追踪到的每个目标与检测到的每个目标之间的 IOU 值，然后使用一个阈值来排除那些 IOU 值较小的对应关系。随后构建一个距离矩阵，并使用匈牙利算法为每个检测到的目标分配一个唯一的 ID。对于那些未能与追踪到的目标匹配的检测结果，将其视为目标消失，并将它们记录在一个消失轨迹的列表中。对于那些检测到的目标，如果它们没有与追踪到的目标匹配，则分配新的 ID，并将其送入追踪模型进行初始化，以便在后续帧中继续进行目标追踪。

Step2：ReID 匹配

由于船只在行进的过程中，出现了部分船只被行进更快的船只完全遮挡之后消失的情况，需要对消失重现的目标定制匹配策略。我们希望对消失的目标与新出现的目标进行外观匹配，故储存了每个目标被送入追踪模型时由 Resnet 提取的模板特征，并为每个建立了特征储存库，这个储存库包含每个目标的多个特征。每个消失的目标计数消失的帧数，当计数达到上限时删除这些消失的目标的轨迹，特别的，本文还为新出现的目标计数出现的帧数，因为在船只再次出现被检测到时，由于被严重遮挡可能无法利用外观特征做出良好的匹配。这时首先为新目标分配新的 id，之后在它出现后的一定帧数里用持续用它的最新特征持续与消失的目标做匹配，具体地说，计算新目标的最新特征与消失的目标的每个特征的欧氏距离并取最小值作为新目标与消失的目标的距离来建立代价矩阵，并设定一个阈值，高于这个阈值的将被排除掉，之后执行匈牙利算法进行匹配。新目标如果匹配到了消失的目标，那么它会被分配到消失的目标的 id，否则会保持新的 id。

3. 实验评估

3.1. 数据集

本文使用的船只数据集由江面上的摄像头云台抓拍提供。原始数据为 1920*1080 分辨率的视频文件。视频的对象为正在航行的船只。由于训练目标追踪器与目标检测器需要不同格式的数据集，我们首先抽取视频片段为每个视频的目标标注位置信息与 ID，获得了 ShipTracker 数据集用来训练 OF-stark 目标追踪模型。ShipTracker 数据集由 16 个视频序列组成，包含 2035 张 1920*1080 分辨率的图像文件。我们使用这些视频文件训练 OF-stark 目标追踪模型。在训练 yolov5 目标检测器时，使用部分 ShipTracker 追踪数据集转化的检测数据集与另外随机抽取的视频帧并标注作为总的检测训练集共记 4052 张 1920*1080 分辨率的图像文件。部分数据集的原始图像如图 5 所示：

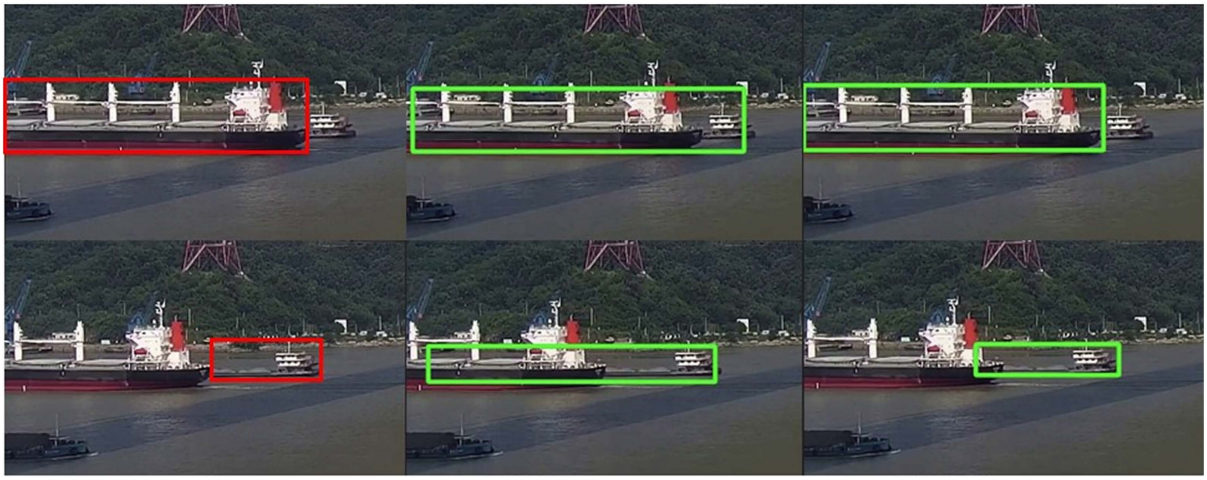
Figure 5. Dataset original images
图 5. 数据集原始图像

3.2. 结果分析

本实验的训练过程在基于 Ubuntu 18.04 操作系统和 CUDA 11.6 的环境下进行。使用一台配备 NVIDIA

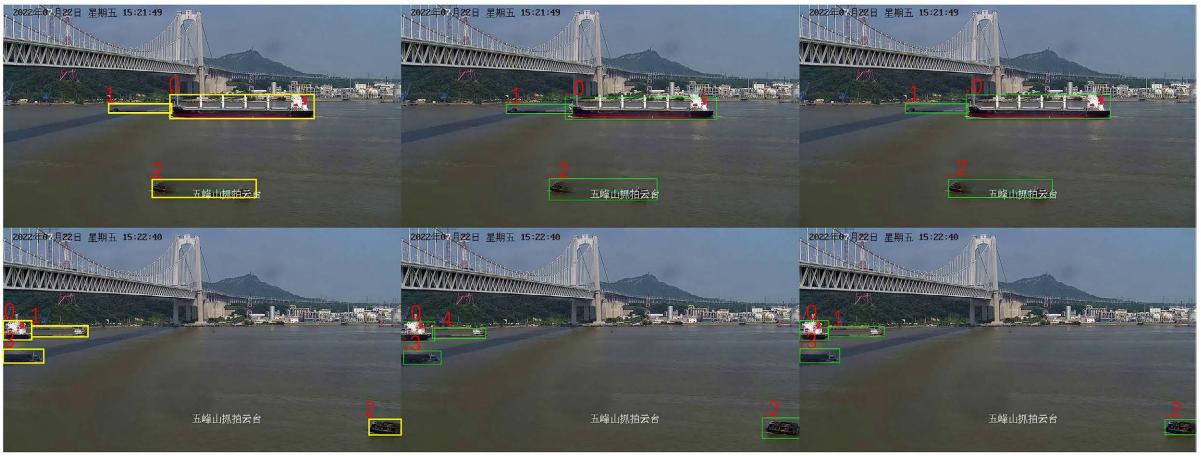
GeForce RTX 3090 GPU (24 GB 显存)的服务器来进行训练。在实验中，只采用 GPU 来进行训练。在训练时，Yolov5 优化器为 ADAM，Batch size 为 8，学习率为 0.0001，训练 200 轮，OF-stark 的学习率为 0.0001，优化器为 ADAMW，Batch size 为 16，首先在 GOT-10K 上训练 500 轮以获得预训练权重，之后再 ShipTracker 数据集上训练 100 轮完成迁移学习。

部分跟踪结果如图 6，图 7 所示。



A. 真实值　　**B. Stark**追踪结果　　**C. OF- stark**追踪结果

Figure 6. Stark (middle) vs. OF-stark (right) tracking accuracy comparison
图 6. Stark (中)与 OF-stark (右)追踪精度对比



A. 真实值　　**B. yolov5 + bytetrack**追踪结果　　**C. yolov5 + OF-stark**追踪结果

Figure 7. YOLOv5 + bytetrack (middle) vs. YOLOv5 + OF-stark (right) tracking results
图 7. YOLOv5 + bytetrack (中)与 YOLOv5 + OF-stark (右)追踪结果

由图 6 可以看出，由于 OF-stark 模型中对光流信息的引入对于追踪的精确性有了显著提高，目标框可以更好的找到目标的边界。图 7 中，A 列图像为一个视频序列中每个船只目标的真实坐标框与正确 id，其中 1 号目标被 0 号目标遮挡消失后在画面中重现，B 列图像为 Bytetrack 方法追踪结果，错误的把消失重现的 1 号目标识别为 4 号新目标。C 列图像为本文的联合多目标追踪框架，1 号目标消失重现后成功的被分配了正确的 id，这说明本文制定的 ReID 策略对目标消失重现后 id 分配的问题有着更好的解决方案。

3.3. 对比试验

对于整个追踪的框架的精准度，本文采用 IDF1 与 MOTP 来评价。IDF1 是度量 ID 被正确分配的指标，其表达式如下：

$$\mathrm{IDF1} = \mathrm{IDTP} / \left(\mathrm{IDTP} + 0.5\mathrm{IDFP} + 0.5\mathrm{IDFN}\right) \tag{11}$$

其中 IDTP，IDFP，IDFN 分别是整个视频中 ID 被正确分配的数量，错误分配的数量和漏分配的数量，MOTP 是衡量跟踪框位置误差的指标，其表达式如下：

$$\mathrm{MOTP} = \sum_{t,i} d_{t,i} \Big/ \sum_{t} c_t \tag{12}$$

其中，$d_{t,i}$ 为第 t 帧第 i 个匹配的定位框与真值的距离，本文使用 IOU 作为距离度量，c_t 是第 t 帧匹配的数目。对比试验结果如下表 1 所示：

Table 1. Comparative experimental results
表 1. 对比实验结果

方法	IDF1	MOTP
Yolov5 + bytetrack	87.2%	0.839
Yolov5 + Deepsort	87.4%	0.824
Yolov5 + OF-stark	**96.7%**	**0.854**

通过对比试验可以看出，本文提出的 Yolov5 + OF-stark 多目标追踪框架无论是在目标 id 准确率还是目标定位准确率上都比其他几种多目标追踪框架更高。

4. 结论

本文提出了一种联合了目标检测和单目标追踪的多目标追踪模型，通过对 Stark 目标追踪模型的改进，获得了更精准的目标定位精度。同时，本文制定了一种简单而有效的目标匹配策略，以缓解目标消失再出现时的 ID 分配问题。

通过定性实验的结果可见，本文的 OF-stark 模型在定位精度方面都取得了显著的提升，这说明利用光流特征对视觉追踪特征进行指导的追踪模型比依赖单帧视觉特征的追踪模型更有效。此外，在船舶追踪任务中，本文的联合多目标追踪框架在 IDF1 指标下比 bytetrack 方法高出 9.5%，比 Deepsort 方法高出 9.3%。在 MOTP 指标下比 bytetrack 方法高 0.015，比 Deepsort 方法高 0.03，这些结果充分说明了本文联合多目标追踪框架具有更高的性能。

在接下来的研究中，我们将继续优化追踪模型，以提高追踪效率并探索更多性能提升的潜力。

参考文献

[1] Bewley, A., Ge, Z., Ott, L., *et al*. (2016) Simple Online and Real-Time Tracking. 2016 *IEEE International Conference on Image Processing* (*ICIP*), Phoenix, 25-28 September 2016, 3464-3468. https://doi.org/10.1109/ICIP.2016.7533003

[2] Wojke, N., Bewley, A. and Paulus, D. (2017) Simple Online and Realtime Tracking with a Deep Association Metric. 2017 *IEEE International Conference on Image Processing* (*ICIP*), Beijing, 17-20 September 2017, 3645-3649. https://doi.org/10.1109/ICIP.2017.8296962

[3] Zhang, Y., Sun, P., Jiang, Y., *et al*. (2022) Bytetrack: Multi-Object Tracking by Associating Every Detection Box. *European Conference on Computer Vision*, Tel Aviv, 23-27 October 2022, 1-21. https://doi.org/10.1007/978-3-031-20047-2_1

[4] Sun, P., Cao, J., Jiang, Y., *et al*. (2020) Transtrack: Multiple Object Tracking with Transformer. https://doi.org/10.48550/arXiv.2012.15460

[5] Meinhardt, T., Kirillov, A., Leal-Taixe, L., *et al*. (2022) Trackformer: Multi-Object Tracking with Transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, 18-24 June 2022, 8844-8854. https://doi.org/10.1109/CVPR52688.2022.00864

[6] Vaquero, L., Brea, V.M. and Mucientes, M. (2023) Real-Time Siamese Multiple Object Tracker with Enhanced Proposals. *Pattern Recognition*, **135**, Article ID: 109141. https://doi.org/10.1016/j.patcog.2022.109141

[7] Cai, J., Xu, M., Li, W., *et al*. (2022) MeMOT: Multi-Object Tracking with Memory. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, 18-24 June 2022, 8090-8100. https://doi.org/10.1109/CVPR52688.2022.00792

[8] Bertinetto, L., Valmadre, J., Henriques, J.F., *et al*. (2016) Fully-Convolutional Siamese Networks for Object Tracking. *Computer Vision ECCV* 2016 *Workshops*, Amsterdam, 8-10 and 15-16 October 2016.

[9] Bhat, G., Danelljan, M., Gool, L.V., *et al*. (2019) Learning Discriminative Model Prediction for Tracking. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, 27 October-2 November 2019, 6182-6191. https://doi.org/10.1109/ICCV.2019.00628

[10] Yan, B., Peng, H., Fu, J., *et al*. (2021) Learning Spatio-Temporal Transformer for Visual Tracking. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, 10-17 October 2021, 10448-10457. https://doi.org/10.1109/ICCV48922.2021.01028

[11] Vaswani, A., Shazeer, N., Parmar, N., *et al*. (2017) Attention Is All You Need. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6000-6010.

[12] Chen, B., Li, P., Bai, L., *et al*. (2022) Backbone Is All Your Need: A Simplified Architecture for Visual Object Tracking. *European Conference on Computer Vision*, Tel Aviv, 23-27 October 2022, 375-392. https://doi.org/10.1007/978-3-031-20047-2_22

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., *et al*. (2020) An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale. https://doi.org/10.48550/arXiv.2010.11929

[14] Xu, H., Zhang, J., Cai, J., *et al*. (2022) Gmflow: Learning Optical Flow via Global Matching. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, 18-24 June 2022, 8121-8130. https://doi.org/10.1109/CVPR52688.2022.00795

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

[16] Carion, N., Massa, F., Synnaeve, G., *et al*. (2020) End-to-End Object Detection with Transformers. *European Conference on Computer Vision*, Glasgow, 23-28 August 2020, 213-229. https://doi.org/10.1007/978-3-030-58452-8_13

[17] Liu, W., Anguelov, D., Erhan, D., *et al*. (2016) Ssd: Single Shot Multibox Detector. *Computer Vision—ECCV* 2016: 14*th European Conference*, Amsterdam, 11-14 October 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[18] Lin, T.Y., Goyal, P., Girshick, R., *et al*. (2017) Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 2980-2988. https://doi.org/10.1109/ICCV.2017.324

[19] Girshick, R., Donahue, J., Darrell, T., *et al*. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

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