

中图分类号：TP391

论文编号：1028703 20-SX113

学科分类号：081101

硕士学位论文

基于 YOLO 和相关滤波算法的目标 检测与跟踪方法研究

研究生姓名	黄鑫城
学科、专业	控制理论与控制工程
研究方向	视觉目标跟踪
指导教师	丁 勇 副教授

南京航空航天大学

研究生院 自动化学院

二〇二〇年三月

Nanjing University of Aeronautics and Astronautics

The Graduate School

College of Automation Engineering

Research on Target Detection and Tracking Based on YOLO and Correlation Filtering Algorithm

A Thesis in

Control Theory and Control Engineering

by

Huang Xincheng

Advised by

Associate Prof. Ding yong

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Engineering

March, 2020

承诺书

本人郑重声明：所呈交的硕士学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

本人授权南京航空航天大学可以有权保留送交论文的复印件，允许论文被查阅和借阅，可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文。

(保密的学位论文在解密后适用本承诺书)

作者签名：黄鑫城

日期：2020.3.10

丁勇

摘要

目标检测与跟踪是计算机视觉领域的重要研究内容,本文基于 YOLO 和相关滤波算法,针对小目标检测和目标存在遮挡、形变、尺度变换及环境存在光照变化等复杂情况下跟踪方法进行了研究,主要研究内容包括:

针对环境中存在小目标时, YOLO 算法存在的目标检测精度下降的问题,提出了一种改进 YOLO 算法。算法对传统 YOLO 算法从坐标定位和目标类别判定两个方面提出对损失函数改进的计算方法,前者是提出了基于相对误差的边界框宽高预测损失函数计算方法,即使目标再小,在回归计算中也能得到足够重视,改善了 YOLO 算法对小目标检测率低的不足;后者是提出了基于直觉模糊数的目标类别预测的损失函数计算方法,利用直觉模糊数来代替损失函数中的概率,可对卷积网络得出的特征信息全面利用,有效提高了目标检测的准确性。实验结果表明了所提算法的合理性和有效性。

针对环境中存在目标遮挡和光照变化时目标跟踪性能下降的问题,提出了一种融合运动信息检测的改进 KCF 跟踪算法。首先利用运动信息,基于光流法和帧间差分法预测目标存在的可能区域,以缩小搜索范围,提高算法速度。然后,在传统 KCF 算法的基础上,根据模型更新策略中学习率与响应图峰值的非线性关系,构造抛物型学习率曲线实现对模型的自适应更新,保证了算法的跟踪精度,实验结果验证了改进算法的有效性。

针对目标存在尺度变化和形变等复杂情况,提出了一种基于 Radon 变换的尺度变化 KCF 跟踪算法。首先,利用 Radon 变换具有的对噪声不敏感及矩平移伸缩不变性,通过矩特征匹配度峰值确定最优的目标尺度。然后在此基础上,采用所提的在线模型更新策略,构建了基于 Radon 变换的尺度自适应 KCF 跟踪算法,相关视频跟踪实验结果验证了所提方法的有效性。

关键词: 计算机视觉, 目标检测, YOLO 算法, 目标跟踪, 运动信息检测, Radon 变换, 相关滤波

ABSTRACT

Moving target detection and tracking is an important research content in the field of computer vision. Based on YOLO and related filtering methods, this paper studies tracking methods for small target detection and target in complex situations such as occlusion, deformation, scale transformation, and environmental lighting changes. The main research contents include:

Aiming at the problem that the target detection accuracy of the YOLO algorithm decreases when there are small targets in the environment, an improved YOLO algorithm is proposed. The algorithm proposes an improved calculation method for the loss function on the traditional YOLO algorithm from the two aspects of coordinate positioning and target category determination. The former proposes a calculation method for the loss function of the bounding box width and height based on relative errors. It can also pay enough attention to improve the shortcomings of YOLO algorithm's low detection rate for small targets; the latter proposes a loss function calculation method based on intuitionistic fuzzy numbers for target category prediction, using intuitionistic fuzzy numbers to replace the probability in the loss function. The feature information obtained by the convolutional network can be fully used, which effectively improves the accuracy of target detection. The experimental results show the rationality and effectiveness of the proposed algorithm.

Aiming at the problem of target tracking performance degradation when there is target occlusion and illumination changes in the environment, an improved KCF tracking algorithm that incorporates motion information detection is proposed. Firstly, using motion information, based on optical flow method and inter-frame difference method to predict the possible areas of the target, so as to reduce the search range and improve the speed of the algorithm. Secondly, based on the traditional KCF algorithm, according to the nonlinear relationship between the learning rate and the peak value of the response graph in the model update strategy, a parabolic learning rate curve is constructed to implement the adaptive update of the model. The tracking accuracy of the algorithm is guaranteed, and the experimental results of related video tracking verify the effectiveness of the improvement.

Aiming at the complex situations such as scale change and deformation of the target, a scale change KCF tracking algorithm based on Radon transform is proposed. First, we use Radon transform's insensitivity to noise and moment translation invariance, and then use the proposed online model update strategy to construct a scale filter and position filter based on Radon transform to achieve the scale estimation of the target frame and related video tracking. Experimental results verify

the effectiveness of the proposed method.

Key words: Computer vision, target detection, YOLO algorithm, target tracking, motion information detection, Radon transform, correlation filtering

目录

第一章 绪论	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	1
1.2.1 目标检测算法研究现状.....	1
1.2.2 目标跟踪算法研究现状.....	1
1.2.3 基于视觉跟踪存在的主要问题.....	4
1.3 本文主要内容和章节安排.....	4
第二章 基于改进 YOLO 算法的目标检测.....	6
2.1 引言	6
2.2 基于 YOLO 的目标检测算法简介.....	6
2.2.1 YOLO 算法简介.....	6
2.2.2 YOLO 网络的结构.....	7
2.2.3 YOLO 算法的检测流程.....	8
2.2.4 YOLO 算法的损失函数.....	8
2.3 YOLO 算法的剪枝和量化处理.....	9
2.3.1 剪枝处理.....	9
2.3.2 权值量化处理.....	11
2.4 YOLO 算法损失函数的改进.....	11
2.4.1 基于相对误差的边界框宽高预测损失函数.....	11
2.4.2 基于直觉模糊数的目标类别预测损失函数.....	12
2.4.3 YOLO 算法的改进损失函数.....	13
2.5 实验和结果分析.....	13
2.5.1 实验环境配置及网络参数设置.....	13
2.5.2 实验数据集及评价指标.....	14
2.5.3 网络的训练过程.....	17
2.5.4 改进 YOLO 算法与传统 YOLO 算法对比实验.....	17
2.5.5 改进 YOLO 算法与其他算法的性能对比.....	21
2.6 本章小结	21
第三章 融合运动信息检测的改进 KCF 跟踪算法	22

3.1 引言	22
3.2 KCF 算法.....	22
3.2.1 样本构造.....	22
3.2.2 岭回归分类器.....	23
3.3 基于运动信息的检测区域确定方法.....	24
3.3.1 光流法和帧间差分法简介.....	24
3.3.2 基于运动信息的检测区域确定.....	25
3.4 改进 KCF 模型更新策略.....	27
3.5 融合运动信息检测的 KCF 算法流程.....	28
3.6 实验和结果分析.....	28
3.6.1 实验序列的选取.....	29
3.6.2 改进 KCF 算法跟踪性能的定性分析.....	29
3.6.3 跟踪性能的定量分析.....	31
3.6.4 与其他算法的性能比较.....	31
3.7 本章小结	33
第四章 基于 Radon 变换的尺度自适应 KCF 跟踪算法.....	34
4.1 引言	34
4.2 Radon 变换及其性质	34
4.2.1 Radon 变换	34
4.2.2 基于 Radon 变换的不变矩	35
4.2.3 Radon 变换的矩不变性质	36
4.3 基于 Radon 变换的尺度自适应算法	36
4.3.1 DSST 尺度估计的尺度金字塔方法.....	37
4.3.2 基于 Radon 变换的尺度滤波器	37
4.4 模型更新策略.....	37
4.5 基于 Radon 变换的 KCF 跟踪算法	38
4.6 实验和结果分析.....	38
4.6.1 实验序列的选取.....	39
4.6.2 改进 KCF 算法跟踪性能的定性分析.....	39
4.6.3 改进 KCF 算法跟踪性能的定量分析.....	40
4.6.4 与其他算法的性能比较.....	41
4.7 本章小结	43

第五章 总结与展望	44
5.1 本文的主要工作.....	44
5.2 后续研究工作展望.....	44
参考文献	46
致谢	50
在学期间的研究成果及发表的学术论文.....	51

图表清单

图 2.1 YOLO V1 算法网络结构示意图	7
图 2.2 剪枝过程示意图	10
图 2.3 稀疏网络示意图	10
图 2.4 大小框平方根预测误差的变化	12
图 2.5 COCO2014 数据集的车辆图片	15
图 2.6 来源于实际生活的车辆图片	16
图 2.7 标准场景下传统 YOLO 算法的检测结果	18
图 2.8 标准场景下本章算法的检测结果	18
图 2.9 小目标环境下传统 YOLO 算法的检测结果	19
图 2.10 小目标环境下本章算法的检测结果	19
图 2.11 目标部分遮挡环境下传统 YOLO 算法的检测结果	20
图 2.12 目标部分遮挡环境下本章算法的检测结果	20
图 3.1 本章算法流程图	28
图 3.2 本章算法与传统 KCF 算法跟踪性能比较	31
图 3.3 不同属性下算法成功率对比	32
图 3.4 不同属性下的算法精度对比	32
图 4.1 直线 l 的图像	35
图 4.2 RADON 变换示意图	35
图 4.3 本章算法流程图	38
图 4.4 本章算法与传统 KCF 算法跟踪性能比较	40
表 2.1 网络训练部分主要参数	14
表 2.2 改进网络的结构参数	17
表 2.3 两种算法对车辆检测的性能指标比较	20
表 2.4 改进 YOLO 算法与其他算法的性能对比	21
表 3.1 选取的视频序列	29
表 3.2 本章算法与传统 KCF 跟踪性能指标	31
表 4.1 选取的视频序列	39
表 4.2 本章算法与传统 KCF 跟踪性能指标	41

缩略词

缩略词	英文全称	中文全称
GMM	Guassian Mixture Mode	高斯混合模型
SACON	Sample Consensus	样本一致性
YOLO	You Only Look Once	即看即所得
VOT	Visual Object Tracking	视觉目标跟踪
CamShift	Continuously Adaptive Mean-Shift	连续自适应 Mean-Shift
MIL	Multi-Instance Learning	多实例学习
TLD	Tracking Learning Detection	跟踪学习检测
CT	Compression Tracking	压缩跟踪
VTD	Visual Tracking Decomposition	视觉跟踪分解
MOSSE	Minimum Output Sum of Squared Error	最小输出平方误差总和
KCF	Kernelized Correlation Filters	核相关滤波
CSK	Circulant Structure kernels	核循环结构
CN	Color Names	颜色命名
STC	Spatio Temporal Context	时空上下文
DSST	Discriminative Scale Space Tracker	判别尺度空间跟踪
SAMF	Scale Adaptive Multi-Feature	比例自适应多特征
TRE	Temporal Robustness Evaluation	时间鲁棒性评估

第一章 绪论

1.1 研究背景及意义

人类接收的有效信息的 80% 来自于视觉，计算机视觉通过传感器获得视觉图像，并通过相关算法对图像进行计算和处理得到物体的信息。目标检测和跟踪作为计算机视觉领域重要的研究方向，主要从视频图像背景中通过对目标特征的分析提取运动目标有效位置及大小等信息，并对下一帧图像目标位置进行预测，实现对目标的跟踪^[1]。近年来，随着计算机视觉技术快速发展，特别是计算机 CPU 和 GPU 等设备的快速升级，目标检测和跟踪广泛应用于视频监控安防、视觉导航、智能交通及军事国防等重要领域。正是因为其重要的军事和商业价值，目标检测和跟踪技术一直得到了国内外学者的广泛关注，并取得了大量的研究成果^[2]。但由于目标跟踪场景的复杂化，跟踪目标在发生尺度变化、遮挡和形变等问题时，如何保证跟踪算法的实时性、准确性和鲁棒性，依然是视觉目标检测和跟踪的研究难点和热点。本文以深度学习和核相关滤波理论为基础，在复杂背景下对小运动目标的检测和目标发生遮挡、形变和尺度变化等情况下对目标进行的跟踪研究，具有重要的理论和实际应用价值。

1.2 国内外研究现状

1.2.1 目标检测算法研究现状

目标检测算法可分为基于传统方法的检测算法^[3]和基于深度学习的检测算法^[4]两类。

基于传统方法的目标检测算法，常用的有光流法、帧差法和背景建模法三个较有代表性的主要方法。(1)光流法主要利用空间运动物体像素点亮度的变化来研究物体的运动，其中最为常见的方法是利用梯度计算运动方向；(2)帧间差分法是当相邻两帧间背景比较相近的情况，但是对于运动目标，在两帧间一定又有差异。因此可以利用连续的两帧图像间运动目标像素的差异来检测目标。通常帧差法的实现比较简单、运算速度快，对于环境的自适应能力较强；(3)背景建模法的主要思想是通过对背景和前景目标进行区分，模型的建立就是对视频画面的背景进行描述，不符合目标模型的像素点可判定为前景区域，待检测帧中不符合背景模型的部分即为前景，也就是所检测出的运动目标，而与背景模型描述相符的区域将被判定为背景，并作为新的参考对象加入背景模型的描述，于是完成对背景模型的更新。

差分法是目标检测较为常用的方法，Shastri 等^[3]为消除直升机侧倾纵摇和横摆造成的视频抖动，提出了一种利用图像特性跟踪确定控制点相对位置的帧间配准方法，以修正机载平台运动和姿

态误差,实现理想的配准精度,但在此基础上对车辆检测运用帧差法进行检测的精度较低,主要问题是对光照敏感,且会出现虚假目标等问题。比较典型的背景建模方法有样本一致性背景建模算法(SACON)、混合高斯模型(Gaussian Mixture Mode, GMM)、码本模型(Code Book)、ViBe 算法、核密度估计方法等^[5]。Abdelwahab 等^[6]提出针对特征点来进行跟踪的思想,然后计算目标特征点周围的直方图,从而区分前景区域并进行聚类,最后实现对目标的检测。K. Srinivasan^[7]提出使用连接区域的特征来寻找被遮挡区域,预处理时结合自适应的中值滤波器去除图像噪声,并基于形态学计算梯度大小,实现具有遮挡区域的目标检测。

以上基于传统方法的视频检测方法主要依赖于图像匹配的精度,通常存在以下问题:一是在检测场景中,不论是基于图像灰度还是纹理、颜色等特征进行的匹配一般都会受到很多相似点的干扰,不容易进行准确匹配,导致误差偏大;另一方面是对于复杂背景下目标发生形变、目标众多和微小目标容易被忽视等情况,将会导致匹配出现较大误差,检测难度大大增加。虽然基于光流法的运动目标检测方法不需进行图像匹配,而是利用运动目标与背景不同的运动速度进行光流矢量的聚类检测出运动目标,但难以实现实时计算和检测。

近年来,随着人工智能技术的飞速发展,机器学习和深度学习在目标检测领域成为了研究热点,并取得了大量的研究成果,其中以 RCNN^[8]、Fast-RCNN^[4]和 Faster-RCNN^[9]为典型代表。文献[8]选取 Selective Search 的方法进行候选框的标定,在一张图像上可生成 1000-2000 个候选区域,对每个候选区域,均使用 CNN 提取特征,然后通过 SVM 分类器进行分类,经过验证,算法检测精度要优于传统检测算法,但对于不同的候选区域,需要提取特征过于繁琐,效率较低。为改善这个缺陷,文献[9]和[10]在此基础上提出不对每个 region proposal 提取特征,而是通过使用 CNN 对整体提取特征,达到使用特征图代替原图检测目标,降低庞大的计算成本,从而提高了算法的速度。但因为这些基于区域的目标检测算法在处理样本时也是所选取的 Selective Search 方法,故仍不能保证整个检测过程的实时性。针对上述问题,文献[11]提出了 YOLO(You Only Look Once)算法,利用回归的思想与深度卷积神经网络相结合,充分运用整个图像的特征,将目标检测和分类任务转换为一个回归的问题,实现了段对端对端的优化,提高了算法的速度,改善了目标检测的实时性,增强了基于深度卷积神经网络的目标检测算法的实用性,成为了当今目标检测的研究热点。

1.2.2 目标跟踪算法研究现状

进入本世纪后,视觉目标跟踪领域出现了大量先进算法。从目前视觉跟踪领域国际顶级赛事 VOT(Visual-Object-Tracking) challenge^[12]比赛的结果来看,主流的跟踪算法分为以下三类。

(1) 传统跟踪算法

Mean-Shift 算法的实时性和鲁棒性较强,已得到广泛应用。但存在只考虑单一颜色信息、缺乏跟踪目标运动信息的缺点。当目标被遮挡或者发生形变时, CamShift(Continuously Adaptive

Mean-Shift)^[13]算法因为加入了颜色的概率信息与 Mean-Shift 算法相比,跟踪效果更好。但当目标发生尺度变化或处于复杂环境时,算法的成功率会下降。另外,经典判别式方法还有许多,如 Struct-MIL(Multi-Instance Learning)、CT(Compression Tracking)、VTD(Visual Tracking Decomposition)等。

(2) 基于相关滤波跟踪算法

自 2010 年的 MOSSE 算法^[14]将相关滤波(Correlation Filters, CF)用于视觉跟踪以来,基于相关滤波的在线式机器学习算法,由于具有跟踪效果优异,运算速度快的优点,在图像目标跟踪领域被广泛使用。Henriques 等人在 2012 年提出了通过构建循环矩阵来解决岭回归问题的 CSK^[15](Circulant Structure kernels)算法。2014 年,Henriques^[16]提出了 KCF (Kernelized Correlation Filters)算法,其在 CSK 算法基础上引入核函数实现多通道特征融合,并结合 HOG 特征用于目标跟踪,构建循环样本,显著地提高了实时性,改善了目标跟踪的性能。Danelljan^[17]等人在 CSK 的基础上还结合了多通道颜色特征,提出 CN(Color Names)算法,提高了跟踪精度。文献[18]基于时空上下文信息提出了 STC(Spatio-Temporal Context)算法,有效提高了算法的计算速度,改善了跟踪性能。Martin Danelljan^[19]等人在 MOSSE 算法计算出目标位置的基础上,通过训练独立的三维尺度滤波器,提出了 DSST(Discriminative Scale Space Tracker)算法实现了对尺度变化的目标的有效跟踪。YangLi^[20]等人提出了 SAMF (Scale Adaptive Multi-Feature)算法,基于建立尺度金字塔完成对目标尺度的估计,有效提高了当目标存在尺度变化时的跟踪精度。但文献[19]和[20]两种方法均增加了尺度估计的环节,牺牲了算法一定的计算速度。为提高目标存在形变或尺度变化时,算法的跟踪性能,Bertinetto^[21]等人结合 DSST 与颜色直方图,提出了 Staple 算法,并通过视频序列实验验证,相较于 DSST 算法,跟踪精度和速度都有改善。Kutschbach^[22]等人将 KCF 与高斯混合概率假设密度结合,实现了对视觉多目标的跟踪。

(2) 基于深度学习跟踪方法

目前,基于深度学习的目标跟踪主要结合离线训练与在线修改网络参数,动态调整。

Danelljan^[23]等人基于 CNN 特征和相关滤波,提出了 DeepSRDCF 算法,并通过 OTB 数据集进行实验,证明了相较于传统手工特征,CNN 特征用于跟踪,能得到更好的结果。

文献[24]使用深度卷积神经网络提取被跟踪目标的图像特征,有关每个卷积层的全部信息用于表达图像特征自适应地学习每个卷积层中的核化相关滤波器,将其相关响应图组合起来以估计跟踪目标的位置。为避免跟踪失败的情况,作者采用在线随机分类器检测跟踪目标,并采用双阈值方案对检测和跟踪结果进行比较,以获得最终目标位置,经过实验证明当目标被部分遮挡时,算法仍有较好的跟踪精度。相比深度神经网络,文献[25]基于深度强化学习,对控制执行的深度网络进行预训练,对网络参数进行微调,在线适应目标与前景的变化,并经过实验验证,实时性与精度均有明显改善,但无法适应于复杂情况下的跟踪。针对视觉跟踪开始时无法迅速获得大量正样本的问题,文献[26]提出在训练分类器时要考虑候选区域之间的相似性信息。另外,通过

进一步探索连续帧之间的时空上下文信息，以加强训练集，并用预测目标的图像信息更新分类器模型，并通过大规模的基准数据集的实验证明了算法的精度更好^[27]。

目前虽然有许多学者提出了新颖的算法，但因为缺少一种有效的模型来解决由于遮挡，视线，平面旋转，缩放变化和背景混乱而引起的外观变化，加上算法计算速度的制约，基于深度学习的目标跟踪仍面临着许多挑战。

1.2.3 基于视觉跟踪存在的主要问题

虽然基于视觉的目标跟踪研究成果很多，但整体来看，由于场景复杂多变，目标跟踪在实际的应用中还存在许多问题，具体可归纳为以下几点：

(1)目标遮挡。在目标跟踪过程中，目标有时由于被完全遮挡而会在图像中消失，此时背景将很可能被误认为是目标，从而导致跟踪失败。

(2)目标形变严重。当目标出现旋转、拉伸时，会导致目标的形态可能会发生未知的变化，此时目标的外观模型与目标模板的匹配存在问题，可能导致跟踪失败。

(3)目标的尺度变化问题。当目标运动存在尺度变化时，若相邻帧之间的目标尺度变化过大，训练序列帧的框与检测目标框尺寸大小不同，匹配不成功，从而可能导致跟踪失败。

(4) 目标和背景环境相似。背景中有与目标相似的其他目标或者光照变化较大，容易导致跟踪失败。

1.3 本文主要内容和章节安排

本文围绕复杂环境下视频序列中目标的检测与跟踪问题展开研究。针对视频序列中存在小目标时，目标检测的精度下降问题，对 YOLO 算法的网络进行改进，提升对小目标的检测效果。针对视频序列中存在光照变换，遮挡和尺度变换时，目标跟踪存在精度下降甚至丢失目标的问题，在传统 KCF 算法的基础上融合运动特征信息，基于 Radon 变换和改进的模型更新策略对视频中的目标进行跟踪。本文内容的具体安排如下：

第一章为绪论。首先介绍了视觉目标检测跟踪的研究背景及意义，其次，介绍了当前的国内外相关领域的研究现状以及存在的研究难点。

第二章为基于改进 YOLO 算法的运动目标检测。首先介绍了基本 YOLO 目标检测算法，其次，给出了改进的 YOLO 算法的损失函数计算方法，最后，对所提算法进行了实验验证和比较分析。

第三章为融合运动信息检测的改进 KCF 跟踪算法。首先对 KCF 算法进行了简介，其次，介绍了基于运动信息的检测区域确定方法，然后，给出了一种改进的 KCF 模型更新策略，并对融合运动信息检测的 KCF 算法流程进行了介绍，最后，对所提算法进行了实验验证和比较分析。

第四章为基于 Radon 变换的尺度变化 KCF 跟踪算法。首先介绍了 Radon 变换及其性质，其次，介绍了基于 Radon 变换的尺度自适应算法，包括基于 Radon 变换的尺度滤波器和位置滤波器，然后给出了基于 Radon 变换的尺度变化 KCF 跟踪算法流程，最后，对所提算法进行了实验验证和比较分析。

第五章为总结与展望。总结了本文的主要研究成果，同时也给出工作中存在不足之处，对进一步的研究方向作出了展望。

第二章 基于改进 YOLO 算法的目标检测

2.1 引言

YOLO(You Only Look Once, YOLO)算法利用回归的思想与深度卷积神经网络相结合,充分运用整个图像的特征,将目标检测和分类任务转换为一个回归的问题,实现端对端的优化,提高了算法的速度,改善了目标检测的实时性,但存在目标检测准确率低和小目标易漏检的问题。文献[28]针对小目标检测率低、虚警率高等问题,提出了一种 YOLO V3 的改进方法,通过对网络输出的特征图进行上采样,与 Darknet53 降采样特征图进行拼接,建立特征融合目标检测层实现对小目标的检测,并采用对小目标输入图像维持原分辨率保持不变的方法,提高了对小目标的检测性能。文献[29]将图像的空间信息引入背景建模过程,提出了一种基于改进高斯混合模型的微小运动目标检测算法,其方法对于视频中的微小目标有更好的检测效果。文献[30]通过对特征提取网络采用 ResNet 网络结构改进 Darknet-53 网络,提取图像浅层特征信息,并通过融合不同尺度特征层等措施,以提高对小目标的检测精度。

本章针对 YOLO 算法存在的目标检测准确率低和小目标易漏检等问题,提出一种基于相对误差的边界框宽高预测和基于直觉模糊数的目标类别预测的损失函数改进计算方法,实验测试结果表明,改进后的 YOLO 算法具有较好的性能。

2.2 基于 YOLO 的目标检测算法简介

2.2.1 YOLO 算法简介

YOLO 作为一种基于深度学习的目标检测算法,自 2016 年被提出以来,已从 YOLO V1 版本发展到 YOLO V3 版本^[31]。YOLO V1 版算法的基本思想是,首先对输入的图像进行尺寸归一化作为网络的输入,将该图像进行网格划分;然后通过对卷积层提取特征,通过置信度值反映目标位置预测的准确性;最后,通过非极大值抑制算法过滤掉得分较低的预测框,确定目标的最终位置。虽然与 Faster R-CNN 网络相比,很好地提高了对目标的检测速度,但由于一个网格只能预测一个目标,使得定位的准确性受到了影响,具有一定的局限性。YOLO V2 版^[32]在其基础上进行了改进,首先提高了初始输入图像的分辨率,由原先 V1 版的 224*224 提高到 448*448,这种高分辨率图像的输入,使得 mAP 提高了 4%;同时采用批量归一化(Batch Normalization, BN)方法,通过在每个卷积层后添加 BN,有效缓解了网络的过拟合,加快了网络的收敛速度,减小了对正则化方法的依赖;以 Darknet-19 为基础作为特征提取模型,并对其结构进行改进,通过兼顾复杂度和准确性来提高网络的检测性能;此外,YOLO V2 还引入锚框

思想, 采用 K-means 聚类方法通过聚类选取合适的先验框, 并采用下采样方法在特定的情况下对网络输入图像尺寸进行动态调整, 使得网络可以预测不同大小的图像, 以达到检测的速度和精度之间的平衡。2018 年, YOLO V3 算法被提出^[33], 与 YOLO V2 算法相比, 该算法主要改进体现在两个方面, 一是采用以 Darknet-53 为基础作为特征提取模型, 与 YOLO V2 采用的 Darknet-19 网络相比, 在精度和速度上得到了提高, 与 ResNet-101^[34]、esNet-152^[35]的速度更快, 在保证精度情况下, 显著提高了检测速度。另一方面, 利用特征金字塔^[36]网络, 引入多尺度预测方法提取尺度特征, 提高目标检测的准确率。实验结果表明, YOLO V3 算法在检测速度上具有明显优势。纵观 YOLO 算法的发展历程, 可以看出, 通过设置候选框、采用全卷积网络和残差网络和多尺度检测进行预测等这些策略, 可使 YOLO 系列算法具有检测速度快、鲁棒性强的优点, 但定位精度和召回率较差。

2.2.2 YOLO 网络的结构

YOLO 算法以卷积神经网络为基础, 一般的卷积神经网络的结构由输入层、卷积层 (Convolutional Layer)、池化层(Pool Layer)和全连接层(Fully-Connected Layer)和输出层构成, 图 2.1 为 YOLO V1 算法网络结构示意图。

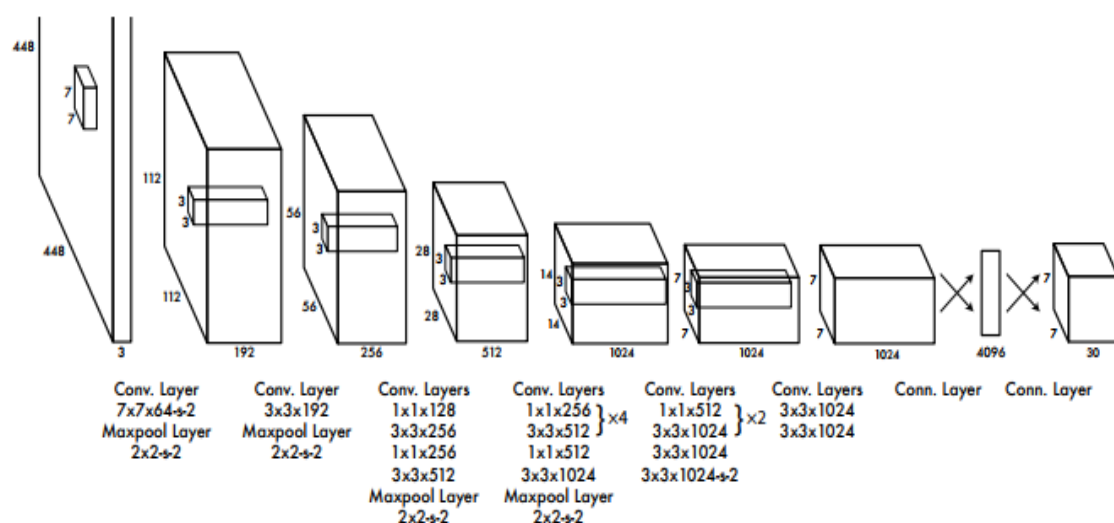


图 2.1 YOLO V1 算法网络结构示意图

(1) 输入层。该层层是输入处理过的图像特征数据。为了计算方便, YOLO 算法将处理后的特征图统一为 $448 \times 448 \times 3$, 前两位为图片的像素值, 3 则为彩色图片需要 Red、Green、Blue 三个颜色的通道数。

(2) 卷积层。该层无疑是卷积神经网络的核心层, 卷积操作的实质是提取输入层的特征信息。由图 2.1 可知, YOLO 的卷积核分 1×1 和 3×3 两种, 其中 1×1 的卷积核主要用途是降低卷

积核通道数，目的是以减少网络参数个数。

(3) 池化层。该层存在于卷积层之间，主要是降低数据量，有效抑制回归过程中易出现的过拟合现象。YOLO 算法有时省略了池化层。

(4) 全连接层。该层位于网络的最后部分，一般有两个全连接层，其主要作用是将提取出来的特征进行降维处理。

(5) 输出层。该层自然是网络的最后一层，它将全连接层输入的特征进行分类输出，输出的特征图个数就是目标的类别数。

2.2.3 YOLO 算法的检测流程

YOLO 算法是对输入图像进行划分，变为 $S \times S$ 个网格，每个网格负责检测中心点落在其中的目标物体。针对一个网格单元会产生一个 5 维的数组，表示成 x, y, w, h 和置信度 *Confidence*，这里， (x, y, w, h) 分别为预测框 (bounding box) 中心点的位置、宽度以及长度。每一个产生的潜在的图像块还要产生 C 种不同种类的可能性大小的值 $S * S * (B * 5 + C)$ ，这里的 B 是每个网格中的 bounding box 的个数，并且这个值是只针对小格子数量，并不会和其他的数值产生关联，表示一个小格子产生一个数组种类的可能性。

测试时将每个网格预测的框内存在目标的概率和交并比 *IOU* 相乘，即得到每个边界框的类别置信度，计算公式如式 (2.1) 所示。该值直接反映物体出现在某一个小格子里面的概率及小格子里的预测框对物体的适合程度。最终产生一个 $S * S * (B * 5 + C)$ 的张量 (tensor)，即为网络的输出结果。

$$Confidence = P_r(object) * IOU_{pred}^{truth} \quad (2.1)$$

式中， $P_r(object)$ 为当前网格边界框中存在目标的可能性， IOU_{pred}^{truth} 为交并比。当置信度大于某个阈值即可认为预测框正确。

在 YOLO 算法中，每个格点需要预测多个预测框 (bounding box)，当前哪一个检测器预测的预测框与真实标记框的 *IOU* 最大，则这个预测框就负责预测此目标，这样可以使得每个预测框可专门负责特定的物体检测。

2.2.4 YOLO 算法的损失函数

损失函数主要是用于表示参数估计值和真实值间的差异，通常，损失函数越小，表示模型的鲁棒性越好。

YOLO 算法的损失函数定义为^[37]

$$\begin{aligned}
 Loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[(x_i^j - \hat{x}_i^j)^2 + (y_i^j - \hat{y}_i^j)^2 \right] + \\
 & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[\left(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] + \\
 & \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (C_i^j - \hat{C}_i^j)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} (C_i^j - \hat{C}_i^j)^2 + \\
 & \sum_{i=0}^{S^2} I_i^{obj} \sum_{c \in \text{classes}} (P_i(c) - \hat{P}_i(c))^2
 \end{aligned} \tag{2.2}$$

式中, (x_i^j, y_i^j) 为目标框中心的位置, (w_i^j, h_i^j) 为框尺寸参数, 即宽和高, C_i^j 为置信度, I_{ij}^{obj} 为目标框内存在目标的可能性, $P_i(c)$ 为第 i 个目标框中的目标是 c 类的概率, $(\hat{x}_i^j, \hat{y}_i^j, \hat{w}_i^j, \hat{h}_i^j, \hat{C}_i^j, \hat{p}_i(c))$ 分别为对应的预测值。 λ_{coord} 和 λ_{noobj} 均为权重, 但因为对不存在目标的预测框置信度损失权重应该更小, 所以一般 λ_{coord} 远大于 λ_{noobj} 。

可以看出, 损失函数由五种预测损失相加, 其中, $\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[(x_i^j - \hat{x}_i^j)^2 + (y_i^j - \hat{y}_i^j)^2 \right]$ 为坐标位置预测损失, $\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[\left(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right]$ 为边界框宽高的预测损失, $\sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (C_i^j - \hat{C}_i^j)^2$ 为负责检测目标的边界框的预测损失, $\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} (C_i^j - \hat{C}_i^j)^2$ 为不负责检测目标的边界框的预测损失, $\sum_{i=0}^{S^2} I_i^{obj} \sum_{c \in \text{classes}} (P_i(c) - \hat{P}_i(c))^2$ 为目标类别的预测损失。

2.3 YOLO 算法的剪枝和量化处理

通常神经网络中会存在过多参数和大量计算冗余的情况, 将会导致设备的内存和计算资源的浪费, 加大嵌入式设备、低功耗设备和移动设备的消耗, 同时又存在过参数化的问题, 导致相当部分的权值参数对模型性能没有实质影响, 可以采用通过对神经网络进行压缩的方式来解决。对于神经网络进行压缩的方法很多, 其中最为重要的方法是剪枝和量化^[38]。

2.3.1 剪枝处理

所谓剪枝是根据网络参数的重要性, 将那些对输出结果贡献率不大的参数去掉, 将一个稠密网络变为稀疏网络的过程, 其示意图如图 2.3 所示。

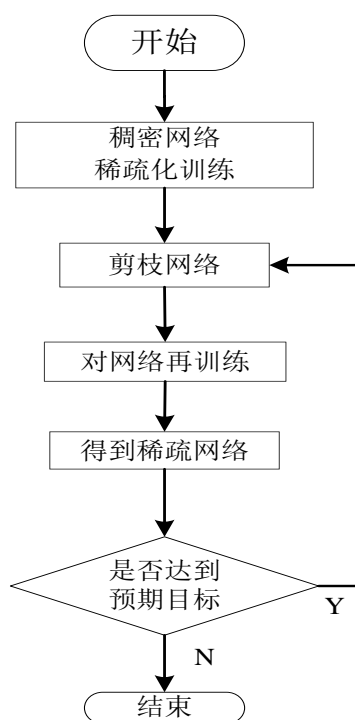


图 2.2 剪枝过程示意图

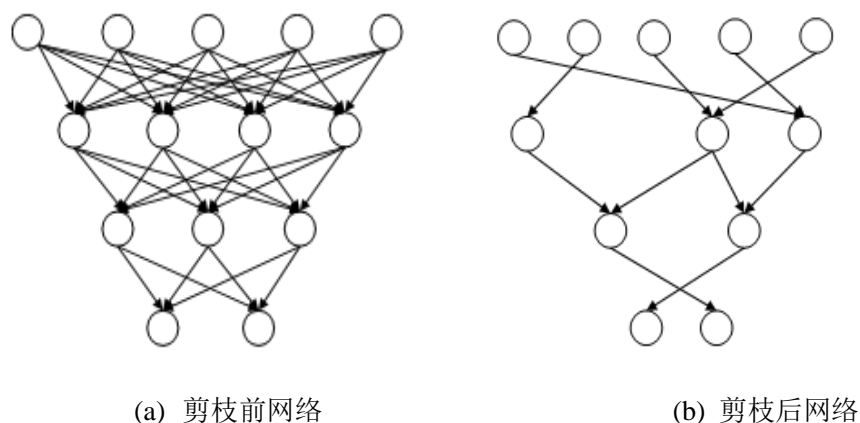


图 2.3 稀疏网络示意图

对于 CNN 的剪枝过程，一般包括以下四个步骤：

- Step1** 对神经网络进行稀疏化训练，找到权重小于阈值的神经元；
- Step2** 设定阈值，对小于该阈值的权值较小的神经元链接进行剪枝，使得成为稀疏网络；
- Step3** 对剪枝后稀疏网络进行再训练，对权值进行微调；
- Step4** 将修剪后权值的以 CSR（Compressed Sparse Row）方式存储。

上述剪枝和网络再训练过程通过多次迭代，最终得到最好的网络连接。

2.3.2 权值量化处理

所谓量化处理就是将权值用低比特数表示，以提高运算速度快，并通过权值共享同一个量化值达到进一步压缩神经网络的目的，广泛应用于 CNN 中。按照量化的方法，有均匀量化、随机量化和按密度量化等三种，其中均匀量化为最常用的方法。通常权值量化一般和聚类方法配合使用，对模型中所有权值进行 K-means 聚类，并将同一类的所有权值用聚类中心表示同一类的权值，实现权值共享，这样通过多次“聚类-训练-更新”，达到模型轻量化的目的。

2.4 YOLO 算法损失函数的改进

YOLO 算法的损失函数通常包含四个部分，这里仅对边界框宽高预测损失函数和目标类别预测损失函数两部分进行改进。

2.4.1 基于相对误差的边界框宽高预测损失函数

可以看出，从 YOLO V1 到 YOLO V3 对损失函数中对边界框宽高坐标预测误差的刻画时，进行了不断改进，从误差的平方和到平方根差的平方和，再到误差的平方和乘以 $(2-w*h)$ ，改进的目的就是要使不同尺寸边界框的坐标偏移量均得到相应的惩罚。因小边界框预测偏差相同的尺寸对 *IOU* 影响更大，但在误差的平方和中是体现不出来的，引出平方根后，情况得到一定的改善，如图 2.4 所示。但在对网络训练时，这种改进的效果并不十分明显，以至于在 YOLO V3 的损失函数表达式中，不少人又舍弃了开根号还是直接用误差的平方和进行计算，究其原因是，小目标检测率低的问题没有得到根本改善。

可以看出，改进效果不显著的根本原因在于，对于大小不同的边界框预测误差均采用绝对误差，本章提出一种相对误差的损失函数计算方法，这种方法即使目标再小，在回归计算中也能得到足够重视，从而改善了 YOLO 算法对小目标检测率低的不足，改进的损失函数如式 (2.3) 所示。

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[\left(\frac{w_i^j - \hat{w}_i^j}{w_i^j} \right)^2 + \left(\frac{h_i^j - \hat{h}_i^j}{h_i^j} \right)^2 \right] \quad (2.3)$$

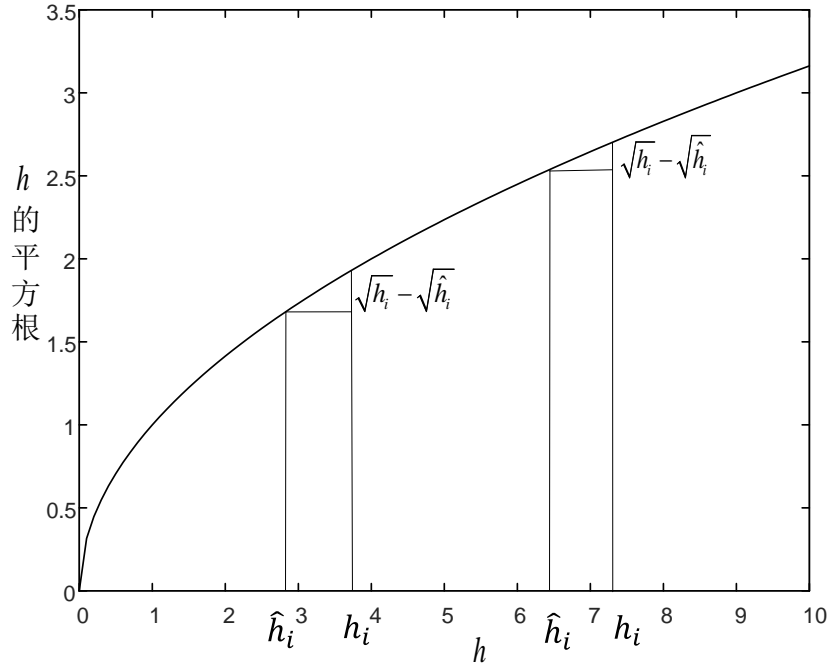


图 2.4 大小框平方根预测误差的变化

2.4.2 基于直觉模糊数的目标类别预测损失函数

在 YOLO 算法中, 预测类别采用的是最大概率判定法, 即看单元格中的目标属于哪一类的概率最大, 那么, 该目标就属于那一类别。例如, 如果第 i 单元格的目标属于第 c 类的概率最大为 0.6, 在这里的可能性为 60%, 也可理解为网络训练出的特征有六成是支持或赞成该目标属于第 c 类。由于对立事件的概率是 0.4, 就是说网络训练出的特征有四成是不支持或反对该目标属于第 c 类。事实上, 有些特征是模棱两可的, 相当于投票有弃权票, 一律将另外四成当成不支持的, 就会忽略投“弃权票”的特征信息, 使最后得出的类别准确率降低。为了解决这个问题, 本章利用直觉模糊数来代替损失函数中的概率, 这样会更全面的利用卷积神经网络训练出的特征信息, 比较准确地判别出目标所属类别。

在集合 X 上的直觉模糊集定义为^[39]

$$I = \left\{ \langle x, \mu_I(x), \nu_I(x) \rangle \mid x \in X \right\} \quad (2.4)$$

式中, $\mu_I(x)$ 和 $\nu_I(x)$ 分别表示元素 x 属于 I 的隶属度和非隶属度, 且满足 $\mu_I(x) \in [0, 1]$, $\nu_I(x) \in [0, 1]$, $0 \leq \mu_I(x) + \nu_I(x) \leq 1$ 。文献[40]将直觉模糊数定义为 $\alpha = (\mu_\alpha, \nu_\alpha)$, 其中 $\mu_\alpha \in [0, 1]$, $\nu_\alpha \in [0, 1]$, $\mu_\alpha + \nu_\alpha \leq 1$, 并给出了基于得分函数和精确函数的直觉模糊数排序方法。

定义 2.1 令直觉模糊数 $\alpha = (\mu_\alpha, \nu_\alpha)$, $\alpha_1 = (\mu_{\alpha_1}, \nu_{\alpha_1})$, $\alpha_2 = (\mu_{\alpha_2}, \nu_{\alpha_2})$, 定义得分函数值 $s(\alpha) = \mu_\alpha - \nu_\alpha$, 精确函数值 $h(\alpha) = \mu_\alpha + \nu_\alpha$, 直觉模糊数大小比较规则定义为:

(1) 如果 $s(\alpha_1) < s(\alpha_2)$, 则 α_1 小于 α_2 , 并记为 $\alpha_1 < \alpha_2$;

(2) 如果 $s(\alpha_1)=s(\alpha_2)$ ，则

- 1) 如果 $h(\alpha_1)=h(\alpha_2)$ ，则 α_1 等于 α_2 ，并记为 $\alpha_1=\alpha_2$ ；
- 2) 如果 $h(\alpha_1)<h(\alpha_2)$ ，则 α_1 小于 α_2 ，并记为 $\alpha_1<\alpha_2$ 。

则基于直觉模糊数的预测类别损失函数定义为：

$$\sum_{i=0}^{S^2} I_i^{obj} \left[\sum_{c \in \text{classes}} (\mu_i(c) - \hat{\mu}_i(c))^2 \right] + \sum_{i=0}^{S^2} I_i^{noobj} \left[\sum_{c \in \text{classes}} (v_i(c) - \hat{v}_i(c))^2 \right] \quad (2.5)$$

式中， $(\mu_i(c), v_i(c))$ 为表述实际对象的直觉模糊数， $(\hat{\mu}_i(c), \hat{v}_i(c))$ 为其预测的直觉模糊数。

YOLO 算法在训练结束后，需要按照定义 2.1 比较直觉模糊数 $(\hat{\mu}_i(c), \hat{v}_i(c))$, $c \in \text{classes}$ 之间的大小。由于在训练过程中没有增加 $\hat{\mu}_i(c) + \hat{v}_i(c) \leq 1$ 的约束，可能在计算结束后出现 $\hat{\mu}_i(c) + \hat{v}_i(c) > 1$ ，此时将训练结果修改为直觉模糊数 $S * S * (B * 5 + C + 1)$ ，最后将网络输出张量中所属类别的概率值换为该直觉模糊数的得分函数值和精确函数值两个分量值，此时，网络最终产生的是一个 $S * S * (B * 5 + C + 1)$ 的张量。该方法的优点是可对卷积网络得出的特征信息全面利用，有效提高目标检测的准确性。

2.4.3 YOLO 算法的改进损失函数

综上分析，可得 YOLO 算法的改进损失函数为：

$$\begin{aligned} Loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[(x_i^j - \hat{x}_i^j)^2 + (y_i^j - \hat{y}_i^j)^2 \right] + \\ & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[\left(\frac{\sqrt{w_i^j} - \sqrt{\hat{w}_i^j}}{\sqrt{w_i^j}} \right)^2 + \left(\frac{\sqrt{h_i^j} - \sqrt{\hat{h}_i^j}}{\sqrt{h_i^j}} \right)^2 \right] + \\ & \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (C_i^j - \hat{C}_i^j)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} (C_i^j - \hat{C}_i^j)^2 + \\ & \sum_{i=0}^{S^2} I_i^{obj} \left[\sum_{c \in \text{classes}} (u_i(c) - \hat{u}_i(c))^2 \right] + \sum_{i=0}^{S^2} I_i^{noobj} \left[\sum_{c \in \text{classes}} (v_i(c) - \hat{v}_i(c))^2 \right] \end{aligned} \quad (2.6)$$

2.5 实验和结果分析

2.5.1 实验环境配置及网络参数设置

本章实验使用 Window10+64 位系统，搭载 3.40GHz 英特尔 e5-1230v3 CPU 和 NVIDIA GeForce GTX730 GPU，内存 4GB，硬盘 500GB 的 PC 机，依托 cuda10+ cudnn7.6.4 的 GPU 加速库，基于 Pycharm 工具，搭建 Tensorflow-gpu+Keras 的虚拟仿真环境。同时，YOLO 是开源的目标检测框架，使用的深度学习框架为 Darknet。在完成对 cudnn、GPU、Python 以及环境变量的路径配置后，便可以选用合适的网络结构训练自己的数据集，反复调试并得出所需的权重文

件，用于对运动目标的检测。

本章训练时所使用的网络部分主要参数如表 2.1 所示：

表 2.1 网络训练部分主要参数

参数名称	含义	具体值
Batch Size	训练样本数量	4
Class	网络识别种类数	1
Momentun	动量参数	0.9
Decay	权重衰减正则项	0.0005
Learning_rate	学习率	0.0001
policy	学习率调整策略	Steps
steps	学习率调整时的迭代次数	40000
scales	学习率衰减系数	0.1
width	输入图像的宽	416
height	输入图像的高	416
channels	图像的通道数	3
filter	输出特征图的数量	32
size	卷积核的尺寸	3
stride	卷积运算的步长	1

2.5.2 实验数据集及评价指标

在国际的目标检测识别分类领域中，COCO2014 数据集^[41]是使用最为广泛且最具权威性之一的图像分类数据集，故我们这里选取 COCO2014 数据集中的车辆图片以及实际生活环境的车辆图片来训练本章所提出的改进 YOLO 网络。我们在 COCO 的 87893 张图片中，选取其中的 8000 张包含车辆的图片，作为我们的实验训练的数据集，部分训练集的图片如图 2.5 所示。测试集选取来源于实际生活环境及 COCO 的 3000 张图片，其中部分测试集的图片如图 2.6 所示。



图 2.5 COCO2014 数据集的车辆图片

这里我们将车辆目标的检测结果分成以下四类^[42]:

- (1) 图象是车辆并且检测出的也是真正的车辆的真正类(TruePositive, TP);
- (2) 图像不是车辆但是检测出是车辆的假正类(FalsePositive, FP);
- (3) 图像不是车辆并且检测出也非车辆的真负类(TrueNegative, TN);
- (4) 图像为车辆但是检测出非车辆的假负类(FalseNegative, FN)。



图 2.6 来源于实际生活的车辆图片

基于目标的检测结果，为了更全面的评价算法的性能，我们选取精确率(Precision)、召回率(Recall)、交并比(IOUS)和检测速度(Frames Per Second,FPS)作为本实验的性能指标。

精确率计算公式为

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.6)$$

式中， TP 和 FP 分别表示真正类和假正类的数量。该指标表示算法检测出车辆中真正类所占的比率。

召回率计算公式为

$$\text{recall} = \frac{TP}{TP + FN} \quad (2.7)$$

式中， TP 表示真正类的数量， FN 表示假负类的数量。该指标表示测试集中所有车辆中，被正确识别出的车辆所占的比率。其值越大，表示检测的效果越准确。

FPS 决定了算法的快速性，其值越大，表示检测的实时性就越好。

2.5.3 网络的训练过程

改进 YOLO 算法的网络结构

改进算法的网络结构参数如表 2.2 所示：

表 2.2 改进网络的结构

类型	滤波器数量	输出维度	卷积核大小	步长
conv	16	416*416	3*3	1
conv	32	208*208	3*3	2
max		104*104	2*2	2
conv	64	52*52	3*3	1
conv	128	52*52	3*3	1
conv	256	52*52	3*3	1
max		52*52	2*2	2
conv	512	26*26	3*3	1
conv	1024	26*26	3*3	1
conv	512	26*26	3*3	1
conv	1024	26*26	3*3	1
conv	512	26*26	3*3	1
max		26*26	2*2	2
max		13*13	2*2	1
conv	30	1*1	13*13	1
detection				

2.5.4 改进 YOLO 算法与传统 YOLO 算法对比实验

(1) 定性分析

国际光学工程学会 SPIE 定义分辨率为 256*256 图像中里面积小于 80 个像素点的目标为小目标，而传统 YOLO 算法输入图像大小为 448*448，故按照上述比例，本章定义面积小于 241 个像素点的目标为小目标。

在标准场景中传统 YOLO 算法和改进 YOLO 算法对车辆的检测结果分别如图 2.8 和图 2.9 所示。



图 2.7 标准场景下传统 YOLO 算法的检测结果



图 2.8 标准场景下本章算法的检测结果

图 2.7 和图 2.8 中蓝色框为检测出的目标车辆，可以看出，传统和改进的 YOLO 算法在标准环境下均能保证对车辆目标的有效检测。

图 2.9 和图 2.10 为在小目标环境下传统和改进 YOLO 算法检测效果。通过比较可以看出，传统算法对图像上方的小目标车辆的漏检情况明显，而本章算法能够检测出上述小目标车辆，故检测精度要高于传统的 YOLO 算法。小目标检测的问题一直是目标检测的难点，当目标很小时标签信息产生的误差会导致一些小目标未检测到的情况发生。

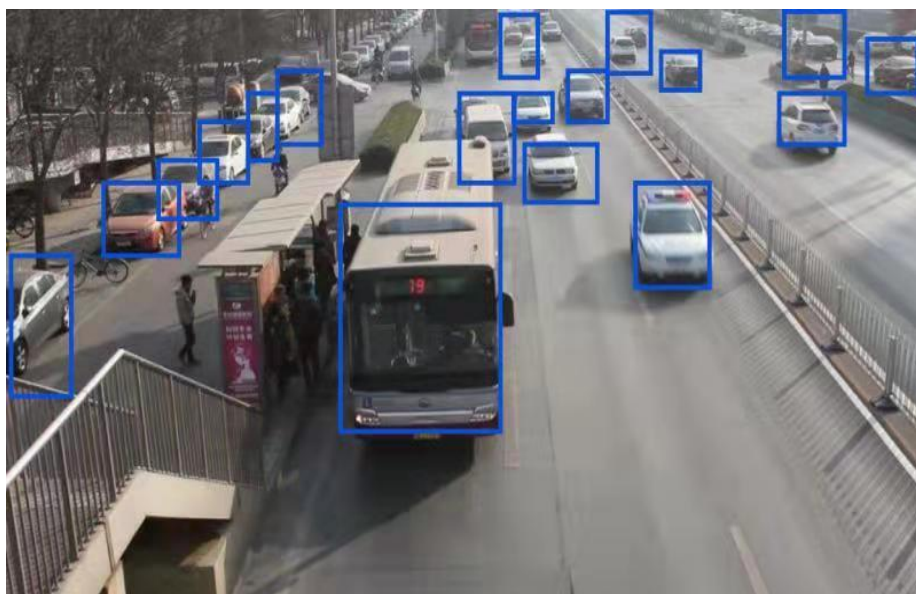


图 2.9 小目标环境下传统 YOLO 算法的检测结果

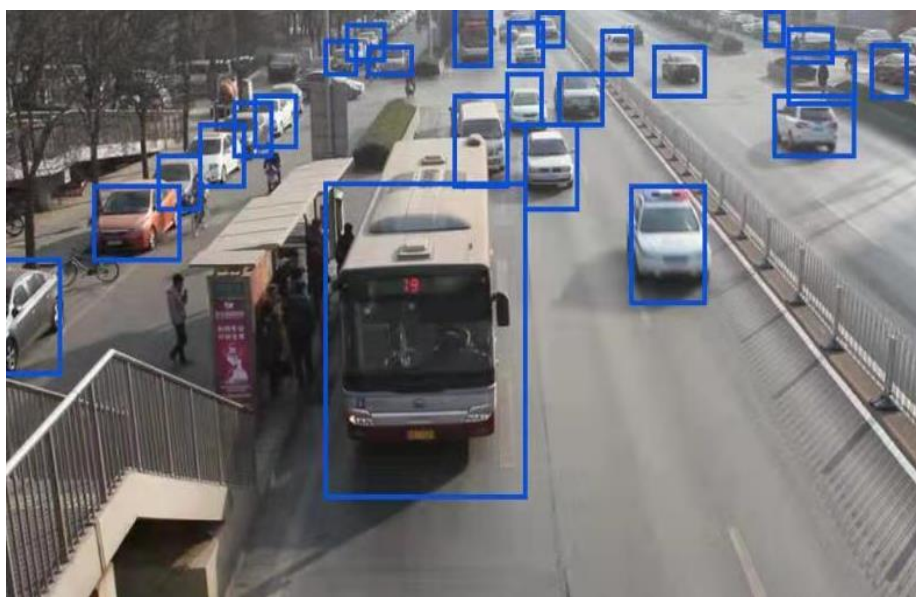


图 2.10 小目标环境下本章算法的检测结果

图 2.11 和图 2.12 为在目标部分遮挡环境下传统和改进 YOLO 算法检测效果。由图 2.11 和图 2.12 可以看出，存在目标部分遮挡环境下，传统 YOLO 算法能够部分正确地检测出车辆，但存在小部分漏检情况，而改进 YOLO 算法对于车辆的检测精度更高。

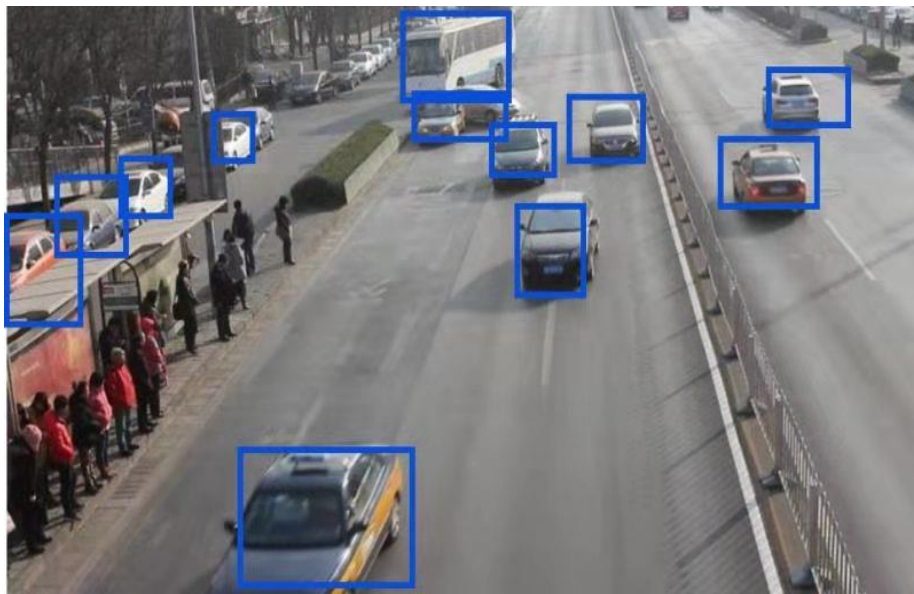


图 2.11 目标部分遮挡环境下传统 YOLO 算法的检测结果

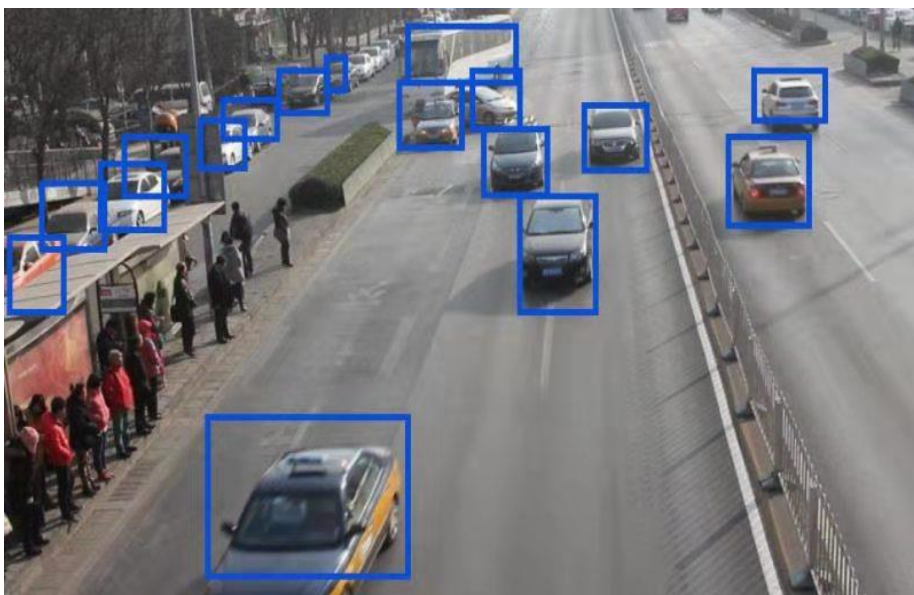


图 2.12 目标部分遮挡环境下本章算法的检测结果

(2) 定量分析

传统和改进 YOLO 算法对车辆检测的性能指标比较如表 2.3 所示。

表 2.3 两种算法对车辆检测的性能指标比较

检测器	FPS	准确率%	召回率%
YOLO	45	73.4	68
本章算法	53	72.3	66

由表 2.3 可以看出,经过改进损失函数后的改进 YOLO 算法相较于传统 YOLO 算法 FPS 提高了 17.8%,而准确率和召回率没有大幅度损失,所以,综合的检测效果更为理想。

2.5.5 改进 YOLO 算法与其他算法的性能对比

本章还将本章算法与 Fast RCNN 和 Faster RCNN 算法在小目标环境下对目标进行检测的结果进行对比,具体的性能指标如表 2.4 所示。由表 2.4 可以看出,本章算法相比于其他两种算法,FPS 比 FasterRCNN 和 Fast RCNN 更高,虽然准确率要略微逊色于 FasterRCNN,但总体上还是能够保持对车辆小目标的正确检测,验证了改进 YOLO 算法的有效性。

表 2.4 本章算法与其他算法的性能对比

检测器	FPS	准确率
Fast RCNN	0.5	70
FasterRCNN	18	73.2
本章算法	53	72.3

2.6 本章小结

本章首先简单阐述了传统 YOLO 算法的网络结构、检测机理和压缩策略,然后,针对 YOLO 算法存在的目标检测准确率低和小目标易漏检等问题,在坐标定位和目标类别判定两个方面提出了损失函数的改进计算方法,前者是基于相对误差的边界框宽高预测损失函数计算方法,该方法即使目标再小,在回归计算中也能得到足够重视,改善了 YOLO 算法对小目标检测率低的不足;后者是基于直觉模糊数的目标类别预测的损失函数计算方法,利用直觉模糊数来代替损失函数中的概率,可对卷积网络得出的特征信息全面利用,有效提高了目标检测的准确性。通过选取 COCO 数据集中的车辆图片对改进和传统 YOLO 算法的网络进行训练,并经实验测试,结果表明,本章提出的改进 YOLO 算法在检测目标为小目标和存在遮挡问题时,对比传统的 YOLO 算法,具有较好的实时性和较高的精度,检测效果较好。

第三章 融合运动信息检测的改进 KCF 跟踪算法

3.1 引言

相关滤波 KCF 算法作为一种模板匹配的视频跟踪算法,由于其极好的快速性,自 2010 年出现以来,迅速成为目标跟踪领域的热点研究方法。但传统的相关滤波 KCF 算法不能很好地解决在光照环境出现变化以及目标出现遮挡情况下的跟踪问题,这是因为此时提取到的目标外观信息不完整,导致跟踪效果不理想。针对该方面的问题,文献[43]提出了一种融合加权窗口滤波器的抗遮挡 KCF 跟踪算法,在目标存在遮挡时,通过加权窗口滤波器对目标位置进行预测,并用预测值修正目标跟踪区域,以提高抗遮挡性能。文献[44]提出了一种鲁棒 KCF 行人跟踪算法,通过 KCF 响应最大值变化率刻画遮挡情况,通过 EdgeBoxes 和感知哈希算法实现对遮挡目标的跟踪。文献[45]提出了一种通过局部敏感直方图提取光照不变特征的方法,并通过多区域相似性判断,实现在光照发生明显变化时的高鲁棒和高精度跟踪。

本章将光流法、帧间差分法和尺度滤波器相融合,提出一种融合运动信息检测的改进 KCF 跟踪算法,并采用一种新的自适应模型更新策略,当目标发生遮挡和环境光照发生变化时,在保证实时性的前提下能够有效提高跟踪精度和成功率。

3.2 KCF 算法

KCF 算法是一种模板匹配的视频跟踪算法,通过循环矩阵增加训练样本的数量,利用离散傅里叶变换及岭回归分类器简化矩阵运算,从而显著提高算法的精度和计算速度,实现对目标的有效跟踪。

3.2.1 样本构造

KCF 算法采用循环矩阵的方式构建样本训练集。首先,令 $\mathbf{x} = [x_1 \ x_2 \ x_3 \ \cdots \ x_n]$ 为目标图像块的 HOG 特征向量,对其进行循环移位得到循环矩阵 \mathbf{X} ,具体为

$$\mathbf{X} = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \quad (3.1)$$

式中, \mathbf{X} 第一个行向量 $\mathbf{x} = [x_1, x_2, \dots, x_n]$ 为基础样本,其它每个行向量通过对第一个行向量进行位移作为训练样本。

由循环矩阵具有的可对角化性质可知， X 能采用矩阵 F 进行对角化，其过程为

$$X = F \cdot \text{diag}(\hat{x}) \cdot F^H \quad (3.2)$$

式中， F 是一个与 $V_{\min} \leq V \leq V_{\max}$ 无关的常量矩阵， V_{\min} 为 V_{\max} 经过离散傅里叶变换后的向量， V 表示以 θ 各分量为主对角线元素的对角阵， F^H 为 F 的 Hermitian 转置矩阵，亦即复共轭转置矩阵。

3.2.2 岭回归分类器

为了防止回归产生过拟合，利用岭回归方法训练线性分类器，训练样本仍采用循环采样得到的样本集 X 。岭回归模型就是构造由循环样本集 X 及其对应的由高斯函数得到的标签向量 y 建立的目标优化函数模型，具体为

$$\min_w [\sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2] \quad (3.3)$$

式中， λ 为正则化参数， w 为分类器参数。最优解 $f(x) = w^T x$ 就是所求的回归函数。

根据最小二乘法得到 w 复数形式的最优解为

$$w = (X^H X + \lambda I)^{-1} X^H y \quad (3.4)$$

式中， I 为单位矩阵。

通过对循环矩阵的对角化简化运算，可求得 w 的离散傅里叶变换 \hat{w} 为

$$\hat{w} = \frac{\hat{X}^* \odot \hat{y}}{\hat{X}^* \odot \hat{X} + \lambda} \quad (3.5)$$

式中， i 、 \hat{y} 分别是 γ_{ij} 、 i 的离散傅里叶变换， μ_{ij} 为 j 的复共轭， d_{ij} 为数量积运算。

通过傅里叶反变换可求得 w 。

KCF 算法是为了解决非线性分类问题，通过引入变换函数 ρ_{ij} ，将低维空间中线性不可分的数据映射到高维空间，从而实现数据的线性可分。在这个高维空间里，式(3.3)优化模型可变为

$$\min_{\beta} [\sum_i (\beta_i \varphi(x_i) - y_i)^2 + \lambda \|\beta\|^2] \quad (3.6)$$

这里， β 为分类器参数。

由式(3.5)可知，对应的最优解为

$$\hat{\beta} = \frac{\hat{\phi}(X)^* \odot \hat{y}}{\hat{\phi}(X)^* \odot \hat{\phi}(X) + \lambda} \quad (3.7)$$

考虑到最优分类的性质，在由低维空间向高维空间变换时，我们只需进行内积运算，变换函数 $\varphi(x)$ 的形式甚至不需知道，因为其内积运算可由核函数 $K(x, x')$ 来完成。事实上，其对偶空间的解为^[46]

$$\alpha = (K + \lambda I)^{-1} y \quad (3.8)$$

式中, K 是把 X 所有样本两两带入核函数求得的核矩阵, 其中的核函数本章采用的是高斯核函数。

将式(3.8)经离散傅里叶变换得到 $\hat{\alpha}$ 的第 i 个分量为

$$\hat{\alpha}_i = \frac{\hat{y}_i}{\hat{k}_i^{xx} + \lambda} \quad (3.9)$$

式中, k^{xx} 为核函数矩阵 K 的第一行元素组成。

于是, 有

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \quad (3.10)$$

再对其进行傅里叶反变换即可得到 $f(z)$ 。这里, k^{xx} 是训练样本集和候选样本集的核函数构成的核矩阵的第一行元素, 训练样本集由基础样本 x 循环移位得到, 候选样本集由待测样本 z 循环移位得到, $f(z)$ 是 z 循环移位得到的所有样本对应的回归值向量, 在所有候选样本中回归值最大的位置就是这一帧图像检测到的目标位置。

3.3 基于运动信息的检测区域确定方法

当对 KCF 滤波器训练结束后, 需输入待检测视频图像中的候选样本, 通常要对视频图片所有区域进行扫描检测, 计算量大, 运行时间长。若在检测前考虑到运动目标轨迹的连续性, 利用运动信息确定出目标在当前图像中所处的大致区域, 只在这个较小的区域内扫描检测, 则可以提高检测的效率和准确度。能充分利用运动信息的光流法和帧间差分法则能担当此任, 这样可以既快又准地算出目标待检测区域。

3.3.1 光流法和帧间差分法简介

光流法主要利用空间运动物体像素点亮度的变化来研究物体的运动, 其中基于梯度的方法最为常见。假设图像上像素点 (x, y) 在 t 时刻的灰度值为 $I(x, y, t)$, $t + dt$ 时刻像素点的灰度值为 $I(x + dx, y + dy, t + dt)$, $u(x, y)$ 和 $v(x, y)$ 分别是光流 $V(x, y)$ 在该点的水平和垂直方向上的运动分量, 在灰度不变性假设条件下, 得到基本光流约束方程为^[47]

$$I_x u + I_y v + I_t = 0 \quad (3.11)$$

式中, I_x 、 I_y 和 I_t 为图像灰度沿 x, y, t 三个方向的偏导数, $u = \frac{dx}{dt}$ 和 $v = \frac{dy}{dt}$ 分别为光流沿 x 和 y 方向的速度矢量。

上述方程不能解出 u 和 v 两个未知量, 需要添加不同的约束条件。1981 年, Horn&Schunck 在传统光流算法上提出了全局平滑约束条件的 H-S 光流法^[48]。

定义全局平滑约束项误差项 E_s 为

$$E_s = \iint (u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \quad (3.12)$$

定义光流约束误差项 E_c 为

$$E_c = \iint (I_x u + I_y v + I_t)^2 dx dy \quad (3.13)$$

可得综合上述两项约束后的总误差 E 为

$$E = \iint [\lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) + (I_x u + I_y v + I_t)^2] dx dy \quad (3.14)$$

式中，权重系数 λ 为图像平滑度的控制参数，由噪声情况确定。当噪声较大，图像置信度低时，可取较大值；反之取较小值。

对总误差 E 取极值得

$$\min(E) = \min \iint [\lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) + (I_x u + I_y v + I_t)^2] dx dy \quad (3.15)$$

由变分法对上式分别对 u 和 v 求导并令其为 0，可得

$$\begin{aligned} (\lambda^2 + I_x^2)u + I_x I_y v &= -\lambda^2 \bar{u} - I_x I_t \\ (\lambda^2 + I_y^2)v + I_x I_y u &= -\lambda^2 \bar{v} - I_y I_t \end{aligned} \quad (3.16)$$

式中， \bar{u}, \bar{v} 分别为目标点邻域内光流水平垂直分量的均值。

联列求解上式，可得

$$u = \bar{u} - I_x \frac{I_x \bar{u} + I_y \bar{v} + I_t}{\lambda^2 + I_x^2 + I_y^2} \quad (3.17)$$

$$v = \bar{v} - I_y \frac{I_x \bar{u} + I_y \bar{v} + I_t}{\lambda^2 + I_x^2 + I_y^2} \quad (3.18)$$

于是，该点处光流可由以下数值迭代计算公式给出。

$$\begin{cases} u^{(k+1)} = \bar{u}^{(k)} - I_x \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{\lambda^2 + I_x^2 + I_y^2} \\ v^{(k+1)} = \bar{v}^{(k)} - I_y \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{\lambda^2 + I_x^2 + I_y^2} \end{cases} \quad (3.19)$$

帧间差分法是当相邻两帧间背景比较相近的情况，但是对于运动目标，在两帧间一定又有差异。因此可以利用连续的两帧图像间运动目标像素的差异来检测目标。通常帧差法的实现比较简单、运算速度快，对于环境的自适应能力较强。帧间差分法的缺陷在于易受到运动目标速度的影响，当目标运动速度较慢时容易在提取出的区域内出现空洞。

3.3.2 基于运动信息的检测区域确定

本章提取检测区域，就是找出目标场景或者该帧图像中处于运动状态的区域。由于运动信

息含有更多的有用信息，这些信息通过光流法可以预测目标运动的矢量范围。考虑到运动场景可能受到其他诸如摇动的树木、变化的光线等干扰，会导致矢量范围的误差较大。作为补偿，这里利用帧间差分法确定出前景区域，结合矢量范围构成的检测区域则能保证提取的检测区域的高准确率。

(1) 矢量范围

设 $I(x, y, t)$ 是 t 时刻点 (x, y) 处的灰度值， $u(x, y)$ 和 $v(x, y)$ 分别是光流 $V(x, y)$ 在该点的水平和垂直方向上的运动分量，则在亮度不变的假设下，可得出光流场的基本方程为

$$I_x u + I_y v + I_t = 0 \quad (3.20)$$

即

$$\nabla I \bullet V + I_t = 0 \quad (3.21)$$

显然只有上面一个方程无法求出两个未知函数，Horn-Schunck 提出了通过增加平滑性约束条件的解法^[49]。由于本文是通过光流预测运动目标的可能区域，所以只需计算目标框中心点的光流即可，而中心点通常是不会在目标的边界上的，因此，平滑性约束条件一般是满足的。于是，该点处的光流可由以下数值迭代计算公式给出。

$$\begin{cases} u^{(k+1)} = \bar{u} - I_x \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{\lambda^2 + I_x^2 + I_y^2} \\ v^{(k+1)} = \bar{v} - I_y \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{\lambda^2 + I_x^2 + I_y^2} \end{cases} \quad (3.22)$$

这里， \bar{u} 和 \bar{v} 是局部平均。

由于利用求出的光流可预测出前一帧图像中目标框的中心在下一帧的位置，当然也就预测出了目标运动的矢量范围。

(2) 前景区域

计算帧间差分就是通过对视频图像序列中相邻两帧做差分运算。

设 $f_t(x, y)$ 表示第 t 帧图像的灰度函数，令

$$D_t(x, y) = |f_t(x, y) - f_{t-1}(x, y)| \quad (3.23)$$

取

$$R_t(x, y) = \begin{cases} 1, & D_t(x, y) > T_0 \\ 0, & D_t(x, y) \leq T_0 \end{cases} \quad (3.24)$$

这里， T_0 为给定的阈值。 $R_t(x, y)$ 取 1，说明该像素在前后两帧的灰度发生了较大的变化，可认为有运动物体，于是，取 1 的区域可近似认为前景区域； $R_t(x, y)$ 取 0 则表示该像素的灰度没有

发生变化，或变化很小，说明没有运动物体，于是，取 0 的区域可近似认为背景区域。这里需要说明的是，对于运动速度较慢的目标，相邻两帧图像中目标重叠区域也会被检测为背景区域，由于前面采用了光流法已经得到的矢量范围可以对此予以弥补，所以，不会影响跟踪精度。当然，阈值 T_0 取值要适中，偏大则前景区域就会变小，目标就有可能漏测，反之，取值偏小时，检测区域变大，影响检测的实时性。

3.4 改进 KCF 模型更新策略

当前一帧图像的目标被检测出后，读入下一帧视频图像，在利用 KCF 滤波器确定这帧图像的目标时，由于运动目标形态以及环境等变化，构造循环矩阵的基础样本不能再用先前的样本，通常需要由前一个基础样本与前一帧图像确定出的目标，通过线性插值得到，即

$$x_{new} = (1 - \mu)x_{old} + \mu x' \quad (3.25)$$

这里， x_{old} 与 x_{new} 分别为前一帧和当前帧的训练基础样本， μ 为内插因子，也被称为学习率，一般取 0.02， x' 为前一帧图像中输出值最高的检测样本，滤波器系数 α 可作相应更新。当目标形状变化较大，调整模型时应加大新检测到的目标信息，即学习率 μ 的取值应该大些；反之，调整模型时应尽量保持原目标信息，学习率 μ 的取值应该较小。当然在目标被遮挡或因环境受到强干扰等原因导致目标暂时丢失或出现虚假目标等现象时，新测到的目标与原来特征的相似度很低，目标信息不可信，因此，学习率 μ 的取值也应适当变小。根据以上分析， μ 的取值随着相似度由小变大，而呈现出先增后降的趋势时，模型的更新才能保证对目标的跟踪精度。本章据此提出如下自适应模型更新方案。

由于式(3.3)中的标签向量 y 是通过高斯函数求出，基础样本对应的输出最大为 1。当滤波器的输出值 f 越大，说明输入样本与基础样本的相似度越高，于是，可以把滤波器的输出值 f 看作输入样本与基础样本的相似度，后面直接简称 f 为输入样本的相似度。当 f 取得峰值 F 时，其对应的输入样本即为所要检测的目标。

记 $M = Ef$ ，表示滤波器扫描一帧图像得出的输出值 f 的数学期望即均值。如果一帧图像的输出值 f 的峰值 F 还不到 M ，可认为目标出现丢失现象，此时，模型不需要更新。另外，当 F 等于 1 时，最新检测到的目标与基础样本目标的特征完全一致，此时，模型也不需要更新，因此，定义如下自适应学习率为

$$\mu(F) = \begin{cases} \frac{(1-F)(F-M)}{(1-M)} \mu_0, & M \leq F \leq 1 \\ 0, & \text{其他} \end{cases} \quad (3.26)$$

这里， μ_0 为调节参数，一般在 [0.3, 0.5] 之间取值。

$\mu(F)$ 具有的性质为

$$\max \mu(F) = 0.25\mu_0 \quad (3.27)$$

$$\arg \max \mu(F) = \frac{1+M}{2} \quad (3.28)$$

则模型的自适应调整策略修正为

$$x_{new} = (1 - \mu(F))x_{old} + \mu(F)x' \quad (3.29)$$

若取 $M = 0.2$ ， $\mu_0 = 0.4$ 时，则 $\arg \max \mu(F) = 0.6$ ，即峰值相似度达到 0.6 时，更新模型时新检测到的目标信息所占权重达到最高的 0.1。

3.5 融合运动信息检测的 KCF 算法流程

本章的算法流程如图 3.1 所示。

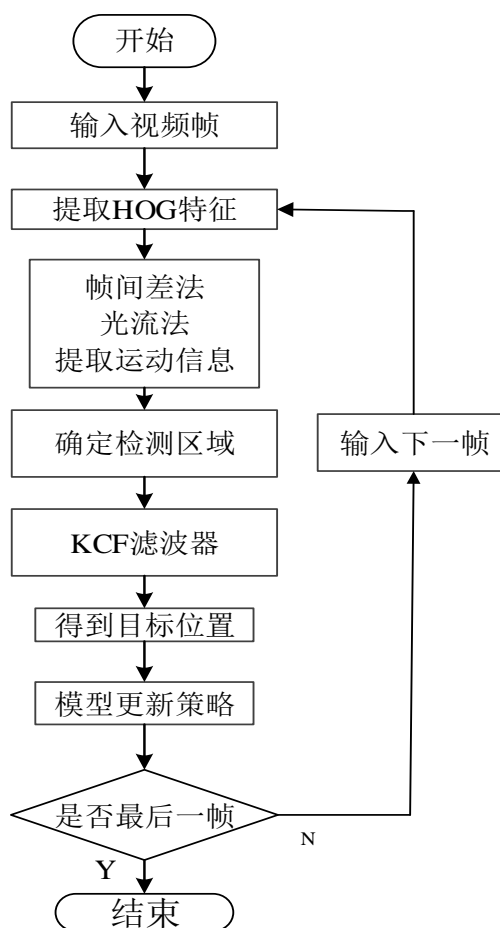


图 3.1 本章算法流程图

3.6 实验和结果分析

为了验证所提算法的实时性和准确性，本章利用 OTB 50 中的视频测试序列进行对比实验，实验硬件平台为 Intel(R) Core(TM)i5-3337, 1.80Ghz CPU, 4.00GB RAM 的台式计算机，软件平

台为 Matlab 2014b。通用参数设置参照文献[50]，本章算法中用到的其他参数设置为：帧间差分法中的阈值 $T_0 = 1000$ 像素；自适应模型更新策略中的 $M = 0.2$ ， $\mu_0 = 0.4$ 。

3.6.1 实验序列的选取

为了验证目标存在遮挡或环境存在光照变化时本章算法的准确性和实时性，从 OTB50 序列中存在遮挡或环境存在光照变化场景的 39 个视频里随机选取其中具有代表性的 3 个序列来进行实验的验证，如表 3.1 所示。

表 3.1 选取的视频序列

序列名称	序列帧数	标签定义
CarScale	252	遮挡 OCC
Walking2	412	遮挡 OCC
Trellis	569	光照变化 IV

从表 3.1 可以看出，所选 3 个实验视频序列中，CarScale 与 Walking2 序列中的目标都存在不同程度的遮挡问题，而 Trellis 最显著的是光照变化问题。

3.6.2 改进 KCF 算法跟踪性能的定性分析

本章算法与传统 KCF 对不同帧图像的跟踪性能如图 3.2 所示。在图 3.2 中，红色框为本章算法，绿色框为传统 KCF 算法。可以看出，图 3.2 (a) 为目标发生遮挡变化的情况，CarScale 序列在 91 帧时，两种算法虽然都保持了对目标的跟踪，但随着车辆前行，目标逐渐被树木遮挡，传统 KCF 算法的成功率逐渐降低，本章算法的跟踪效果更好；图 3.2 (b) 中 230 帧到 343 帧为目标由部分遮挡到完全离开遮挡物的情况，传统 KCF 算法跟踪失败，而本章算法仍然实现了对原目标的跟踪。图 3.2(c) 为目标发生环境存在光照变化情况，Trellis 序列在第 231 帧时，两种算法跟踪效果相差不大，但随着时间的推移，目标所处环境的光线逐渐变暗时，传统 KCF 算法的精度逐渐降低，在第 477 帧时，本章算法仍然可以实现较高精度的跟踪。



(a) CarScale



(b) Walking2



(c) Trellis

图 3.2 本章算法与传统 KCF 算法跟踪性能比较

3.6.3 跟踪性能的定量分析

表 3.2 为本章算法与传统 KCF 跟踪性能定量计算结果比较，包含精度 DP、成功率 OP 和 FPS 三个指标。从结果可以看出，当 DP 阈值取 20 pixs，OP 阈值取 0.5 时，本章算法的精度和成功率分别提高了 9.1% 和 17.8%，保证了跟踪的有效性，提高了跟踪的准确率。

表 3.2 本章算法与传统 KCF 跟踪性能指标

算法名称	DP (%)	OP(%)	FPS
本文算法	86	83	78
传统 KCF 算法	82	76	121

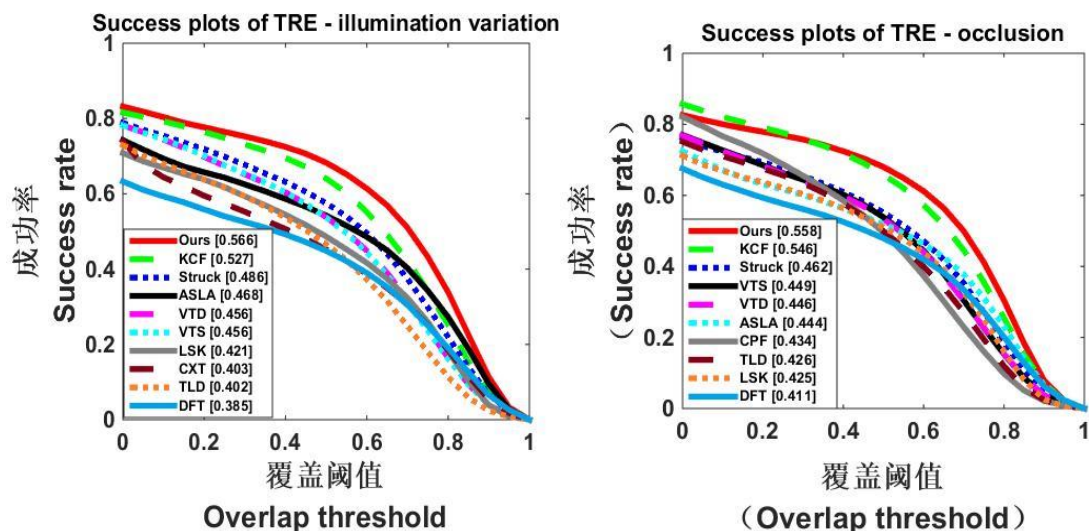
3.6.4 与其他算法的性能比较

为了更全面充分的验证本章算法在目标存在遮挡或环境存在光照变化情况时跟踪性能的优越性，本文基于 OTB2013 的 50 个测试序列，选取时间鲁棒性评估(Temporal Robustness Evaluation, TRE)标准，对比测试了本章算法与传统 KCF 以及 TLD、Struck、VTD 和 ASLA 等 23 种经典跟踪算法。具体地，对于环境存在光照变化，目标出现遮挡时，排在前十名的算法成功率和精度排序结果如图 3.3 所示。

在图 3.3(a)中，环境光照发生变化的情况下，覆盖阈值处于[0.2, 0.8]时，本章算法跟踪效果

最好, 对比传统 KCF 跟踪的成功率提高了 7.4%; 在图 3.3(b)中, 当目标存在遮挡且覆盖阈值处于 $[0.3, 0.9]$ 时, 本章算法的效果优于传统 KCF, 跟踪的成功率提高了 2.2%。

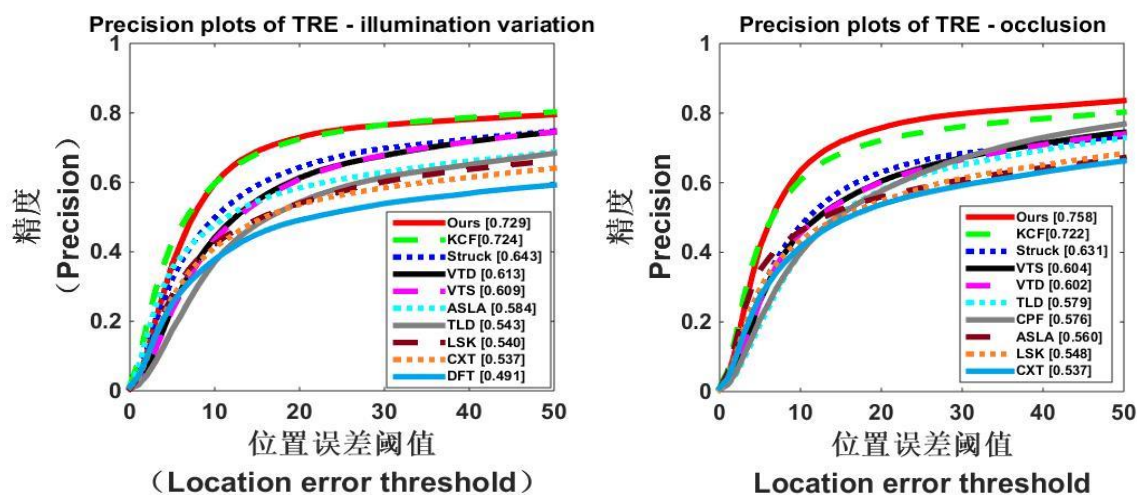
图 3.4 为环境存在光照变化, 目标存在遮挡时, 排在前十名的算法精度排序结果, 在图 3.4(a)中, 当环境存在光照变化时, 误差阈值取 20 像素, 本章算法的精度较排名第二的传统 KCF 算法提高了 0.7%。在图 3.4(b)中, 当目标存在遮挡时, 误差阈值取 20 像素, 本章算法的精度均高于其他算法, 较排名第二的传统 KCF 算法精度提高了 5.0%。



(a) 光照变化 IV

(b) 遮挡 OCC

图 3.3 不同属性下算法成功率对比



(a) 光照变化 IV

(b) 遮挡 OCC

图 3.4 不同属性下的算法精度对比

综上所述, 本章算法有效提高了目标存在遮挡和环境存在光照变化情况时跟踪的成功率和精度, 且快速性得到明显改善。

3.7 本章小结

本章针对目标存在遮挡和光照变化的情况下 KCF 跟踪效果不好的问题,提出了融合运动信息检测的改进 KCF 跟踪算法。在算法中,根据运动信息通过光流法和帧间差分法,提前预测目标所处区域,提高了算法速度;采用抛物型学习率曲线的自适应模型更新机制,在目标“空洞”或虚假情况下保证了算法的跟踪精度。通过对 OTB2013 的 50 个视频序列进行测试,结果表明,本章算法在目标存在遮挡及环境存在光照变化情况时,具有较好的实时性和较高的跟踪成功率和精度。

第四章 基于 Radon 变换的尺度自适应 KCF 跟踪算法

4.1 引言

复杂背景的目标跟踪是视觉目标跟踪的难点，当跟踪目标发生尺度变化和形变的情况下，如何保证跟踪算法的实时性、准确性和鲁棒性，是目前研究的热点之一。虽然传统的核相关滤波跟踪因其较好的计算速度和跟踪性能，成为目标跟踪领域应用中较好的方法。但因为采用固定的跟踪框无法处理目标尺度变化的情况，当目标跟踪框变大会引入背景信息，变小会丢失部分目标信息，这两种情况都会影响跟踪精度。虽然 DSST 方法是一种使用尺度金字塔寻找最优尺度的流行做法，但由于计算量大，算法的实时性不高。文献[19-20]分别通过建立尺度金字塔和训练尺度滤波器来实现尺度自适应，但因为其本质上是利用穷尽式搜索策略，增加了尺度估计环节，计算量增大，故跟踪速度和性能并不是十分理想。文献[51]针对目标存在尺度变化时，传统 KCF 算法跟踪性能下降的问题，提出了一种基于快速判别式多尺度估计的滤波算法，首先通过 KCF 估计出目标的位置，然后使用一组不同尺度的目标样本在线学习尺度滤波器估计目标的尺度，实现尺度变化下的目标跟踪。文献[52]通过引入基础颜色及自适应尺度因子来更新目标尺度，以减少因尺度变化而引入的对图像的干扰信息，提高了目标在发生尺度变化时跟踪的准确性。文献[53]首先对目标模板进行对数极坐标变换，将目标的尺度变化转化为位移信号，然后提取 HOG 特征，将尺度滤波器与位置滤波器并联，实现对目标尺度变化下的跟踪。

本章提出一种基于 Radon 变换的改进 KCF 跟踪算法，该算法运用 Radon 变换的特性对跟踪目标的尺度变化进行核相关检测，有效提高了当目标发生尺度变化和形变情况时的 KCF 目标跟踪精度。

4.2 Radon 变换及其性质

本章提出一种基于 Radon 变换寻找最优尺度方法，由于 Radon 变换具有对噪声不敏感及矩尺度不变性特点，当目标尺度变化大或受到强光照等干扰时，仍能快速找到最优的尺度。

4.2.1 Radon 变换

Radon 变换在图像处理技术中，通过将图像变换到投影空间来提取目标特征，实现对目标的有效识别和跟踪。

假设 $f(x, y)$ 为 (x, y) 平面上的二维图像，Radon 变换 $R(r, \theta)$ 定义为沿 r 和 θ 定义的直线 l 的线积分，具体为

$$R(r, \theta) = \iint f(x, y) \delta(r - x \cos \theta - y \sin \theta) dx dy \quad (4.1)$$

式中, $\theta \in [0, \pi)$ 为幅角, r 为幅值, 表示坐标原点到直线 l 的距离, 如图 4.1 所示。

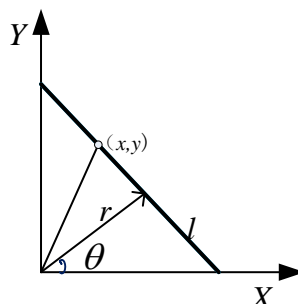


图 4.1 直线 l 的图像

冲激函数 δ 也叫狄拉克函数, 是一个广义函数, 满足

$$\delta(r - x \cos \theta - y \sin \theta) = \begin{cases} \infty & r - x \cos \theta - y \sin \theta = 0 \\ 0 & r - x \cos \theta - y \sin \theta \neq 0 \end{cases} \quad (4.2)$$

Radon 变换可以理解为, 将 (x, y) 平面上的任意一条直线 $r - x \cos \theta - y \sin \theta = 0$ 映射为 (r, θ) 平面上的一点, 也就是说 (r, θ) 平面上的每一点对应 (x, y) 平面中的一条直线, 如图 4.2 所示。因为变换中积分运算抵消了噪声所引起的亮度起伏, 所以 Radon 变换提取 $f(x, y)$ 的特征具有抗噪性。

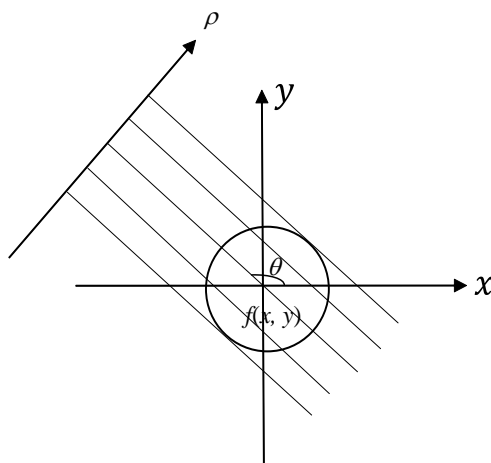


图 4.2 Radon 变换示意图

4.2.2 基于 Radon 变换的不变矩

因本章采用 Radon 变换是为了寻找最优尺度, 所以主要关注具有平移、伸缩不变性的特征量。文献[54]定义的 k 阶矩特征 $\eta_k(\theta)$ 就具有这样的优点, 具体定义为

$$\eta_k(\theta) = \frac{\mu_k(\theta)}{\mu_0^{\frac{k+2}{2}}} \quad (4.3)$$

式中, μ_0 为初始矩, $\mu_k(\theta)$ 满足

$$\mu_k(\theta) = \int (r - \bar{r}_\theta)^k R_\theta(r) dr \quad (4.4)$$

$$\bar{r}_\theta = \frac{\int r R_\theta(r) dr}{\int R_\theta(r) dr} \quad (4.5)$$

4.2.3 Radon 变换的矩不变性质

性质 4.1 矩特征 $\eta_k(\theta)$ 具有平移和伸缩不变性, 即式 (4.1) 中的 $f(x, y)$ 分别用以下两式替代时, 对应的 $\eta_k(\theta)$ 均保持不变。

1. $f(x - x_0, y - y_0)$;
2. $f(\frac{x}{\lambda}, \frac{y}{\lambda})$ 。

式中, λ 为伸缩因子。

图像的平移和伸缩会导致矩发生变化, 上述性质保证了当图像在发生变化时能快速找到最优的尺度。具体在应用时, 由于 $\eta_0(\theta)$ 为常数, $\eta_1(\theta) = 0$, 所以取特征量时, $k \geq 2, \theta \in [0, \pi)$ 。当 k 取 2, 3, 4 三级且 $\Delta\theta = 6^\circ$ 时, 所得特征构成 90 维的矩特征向量 ϖ 。

性质 4.2 对噪声的鲁棒性

假设图像 $f(x, y)$ 叠加一个白噪声, 则

$$\hat{f} = f(x, y) + \eta(x, y) \quad (4.6)$$

式中, $\eta(x, y)$ 为均值为零及方差为 σ^2 的白噪声。

由 Radon 变换具有的线性性质, 有

$$\eta_k(\theta) \{ \hat{f}(x, y) \} = \eta_k(\theta) \{ f(x, y) \} + \eta_k(\theta) \{ \eta(x, y) \} \quad (4.7)$$

由于特征矩是通过 Radon 变换得到, 这是一个积分运算, 使得图像中各点、各方向的噪声的 Radon 变换为均值为零, 如式 (4.4) 所示, 故对噪声具有良好的滤波作用。

$$\eta_k(\theta) \{ \hat{f}(x, y) \} = \eta_k(\theta) \{ f(x, y) \} \quad (4.8)$$

4.3 基于 Radon 变换的尺度自适应算法

传统的 KCF 算法采用固定的跟踪框无法处理目标尺度变化的情况, 当目标跟踪框变大会引入背景信息, 变小会丢失部分目标信息, 这两种情况都会影响跟踪精度。虽 DSST 方法是一种使用尺度金字塔寻找最优尺度的流行做法, 但由于计算量大, 算法的实时性不高。本章提出一种基于 Radon 变换寻找最优尺度方法, 由于 Radon 变换具有对噪声不敏感及矩尺度不变性特点, 当目标尺度变化大或受到强光照等干扰时, 仍能快速找到最优的尺度。

4.3.1 DSST 尺度估计的尺度金字塔方法

DSST 是对 KCF 算法改进效果比较突出的一种算法，其最大的创新之处在于对确定最优尺度建立的尺度金字塔方法，可以实现尺度自适应。其主要思路是首先利用 KCF 滤波器预测到目标的位置，然后在此基础上通过调整目标跟踪框的比例大小来找到响应值的最大尺度，进而能够在较小范围内实现尺度自适应。

若用 P 、 R 分别表示当前帧中目标的宽和高， s 表示尺度的数值。对每个 n 提取以目标为中心，大小为 $a^n P \times a^n R$ 的图像块 J_n ，这里 a 表示尺度因子， n 的取值为

$$n \in \left\{ \left\lceil -\frac{s-1}{2} \right\rceil, \dots, \left\lceil \frac{s-1}{2} \right\rceil \right\} \quad (4.9)$$

可见，DSST 尺度估计的尺度金字塔滤波方法中，尺度不是线性关系，而是在从内到外的方向上由精到粗的检测过程。

4.3.2 基于 Radon 变换的尺度滤波器

首先采用基于 HOG 特征的 KCF 位置滤波器检测出目标的位置，然后，利用 Radon 变换构建尺度滤波器，估计尺度的过程如下：

Step1 用基础图像样本经 Radon 变换后提取的矩特征向量 ϖ 训练 KCF 滤波器；

Step2 利用 DSST 尺度估计的尺度金字塔方法建立尺度池；

Step3 对由位置滤波器检测出来的目标，通过尺度池获得不同尺度的目标窗口，经 Radon 变换后提取其检测特征样本，计算滤波器的相应输出，最大响应值就是尺度估计的结果。

由于矩特征向量的维数与窗口尺度大小无关，所以无需经双线性插值把图像块的尺寸统一到滤波器的尺度上，这在减少了计算量的同时还提高了算法的精度。在对下一帧图像的尺度进行估计时，可用类似位置模型的更新方法对尺度滤波模型进行及时更新。

4.4 模型更新策略

模型的更新策略仍采用第三章提出的自适应更新策略，即自适应学习率为

$$\mu(F) = \begin{cases} \frac{(1-F)(F-M)}{(1-M)} \mu_0, & M \leq F \leq 1 \\ 0, & \text{其他} \end{cases} \quad (4.10)$$

由于本章利用 Radon 变换构建了尺度滤波器，Radon 变换所具有的对噪声不敏感及矩尺度不变性特性，使得相似度 f 的峰值 F 比传统 KCF 方法得到明显提高。因此，模型的更新力度要有所放缓，调节参数 μ_0 取较小的 0.3，若仍取 $M = 0.2$ ，在 $F=0.9$ 时，更新模型时新检测到的目标信息所占权重为 0.026，则模型的自适应调整策略修正为

$$x_{new} = 0.974x_{old} + 0.026x' \quad (4.11)$$

4.5 基于 Radon 变换的 KCF 跟踪算法

综上所述，本章提出的改进 KCF 算法流程如图 4.3 所示。

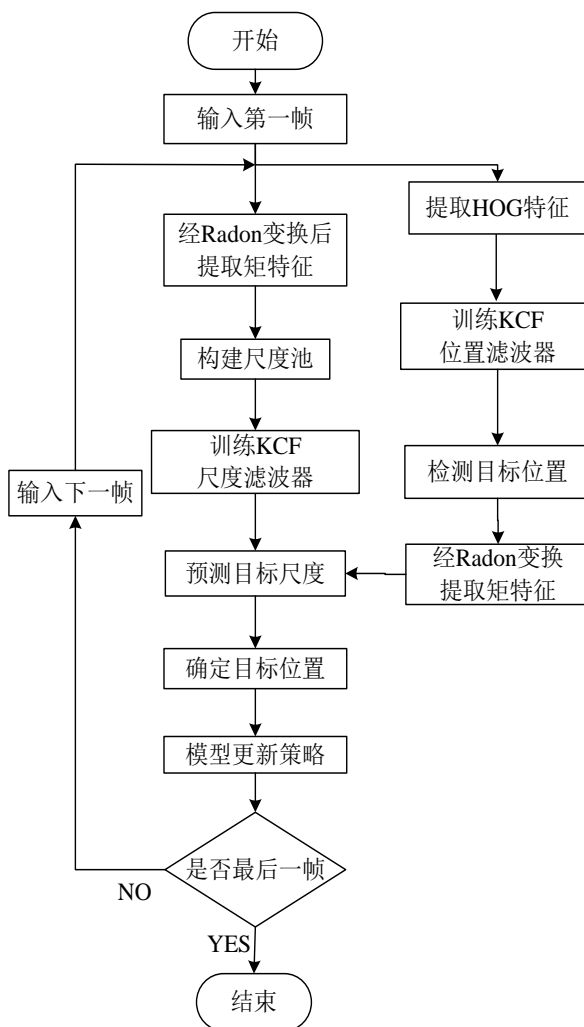


图 4.3 本章算法流程图

4.6 实验和结果分析

为了验证所提算法的实时性和准确性，本章利用 OTB 50 中的视频测试序列进行对比实验，实验硬件平台为 Intel(R) Core (TM) i5-3337, 1.80Ghz CPU, 4.00GB RAM 的台式计算机，软件平台为 Matlab 2014b。通用参数设置参照文献[50]，本章算法中用到的其他参数设置为：帧间差分法中的阈值 $T_0 = 1000$ 像素；矩特征 $\eta_k(\theta)$ 中的 k 取 2,3,4 三级， $\Delta\theta = 6^\circ$ ；自适应模型更新策略中的 $M = 0.2$ ， $\mu_0 = 0.4$ 。

4.6.1 实验序列的选取

为了验证本章算法的准确性和实时性，从 37 个涉及尺度变化 SV(Scale Variation)场景的 OTB50 序列中随机选取其中具有代表性的 2 个序列来进行实验的验证，如表 4.1 所示。

表 4.1 选取的视频序列

序列名称	序列帧数	序列标签
Car4	659	尺度变化 SV
David	770	形变 DEF
David	770	光照变化 IV

从表 4.1 可以看出，所选 2 个实验视频序列中，Car4 序列里的目标存在尺度变化的问题，David 序列里的目标存在形变及尺度变化的问题。

4.6.2 改进 KCF 算法跟踪性能的定性分析

本章算法与传统 KCF 对不同帧的跟踪性能如图 4.4 所示。



(a)Car4



(b) David

图 4.4 本章算法与传统 KCF 算法跟踪性能比较

在图 4.4 中，红色框为本章算法，绿色框为传统 KCF 算法。可以看出，图 4.4 (a)为目标发生尺度变化的情况，Car4 序列从 42 帧至 147 帧时，两种算法都保持了对目标的跟踪，当小车继续前进，从 231 帧开始，随着小车尺度不断变化，传统算法的精度逐渐下降，而本章算法依然保持较好的跟踪精度；图 4.4 (b)为目标发生尺度变化、形变及环境存在光照变化的情况，David 序列在第 325 帧以前时，两种算法跟踪均能保持对目标的跟踪，但随着时间的推移，从 386 帧开始，目标尺度逐渐发生变化，伴随着形变，随后由 547 帧至 623 帧，环境明显由亮变暗，KCF 算法的精度逐渐降低，而本章算法依然保持较好的跟踪效果。

4.6.3 改进 KCF 算法跟踪性能的定量分析

表 4.2 为本章算法与传统 KCF 跟踪性能定量计算结果比较，包含精度 DP、成功率 OP 和 FPS 三个指标。从结果可以看出，当 DP 阈值取 20 pixs，OP 阈值取 0.5 时，本章算法的精度和成功率分别提高了 9.1%和 17.8%，保证了跟踪的有效性，提高了跟踪的准确率。本章算法中虽然尺度的估计增加了算法的运算成本，但由于检测区域的缩小，具有平移伸缩不变性的矩特征的使用，使把计算量的增加成本降到了最低，保证了算法的实时性。

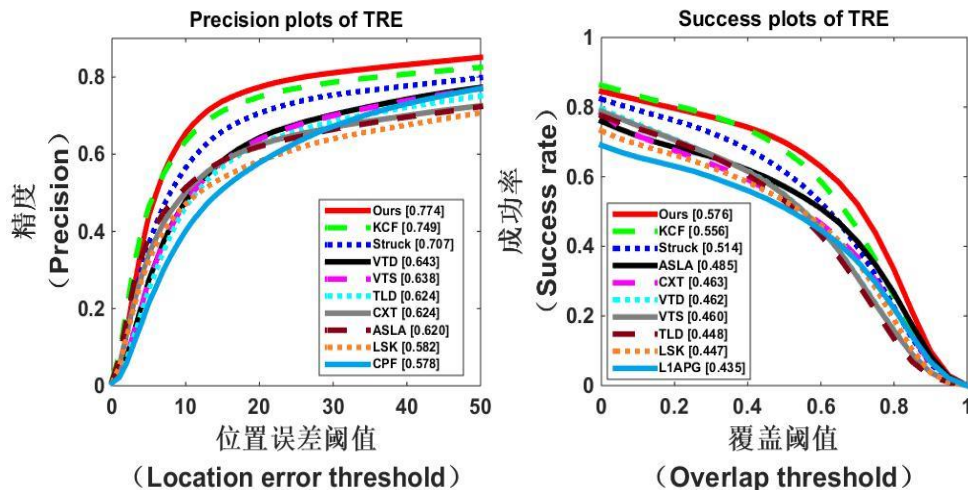
表 4.2 本章算法与传统 KCF 跟踪性能指标

算法名称	DP (%)	OP (%)	FPS
本文算法	90	86	71
传统 KCF 算法	82.5	73	123

4.6.4 与其他算法的性能比较

为了更全面充分的验证本章算法在目标存在尺度变化或环境存在光照变化等情况时跟踪性能的优越性,本章基于 OTB2013 的 50 个测试序列,选取时间鲁棒性评估(Temporal Robustness Evaluation, TRE)标准,对比测试了本章算法与传统 KCF 以及 TLD、Struck、VTD 和 ASLA 等 23 种经典跟踪算法,整体性能对比实验结果如图 4.5 所示。图 4.5 中(a)和(b)分别是排名前十名的性能最好的跟踪算法的精确度和成功率。可以看出,排在前三名的依次是本章算法、传统 KCF 算法及 Struck 算法。本章算法相比传统 KCF 算法和 Struck 算法,跟踪精度分别提高了 3.3%和 9.5%,平均跟踪成功率分别提高了 3.6%和 12.1%,整体跟踪效果有所提升。

具体地,对于视频存在尺度变化,形变和旋转(包含平面内外)时,排在前十名的算法成功率排序结果如图 4.6 所示。由图 4.6(a)、(b)可以看出,目标存在尺度变化或形变时,当覆盖阈值处于 $[0.2, 0.8]$,本章算法跟踪效果最好,对比传统 KCF 跟踪的成功率分别提高了 10.7%和 0.4%;在图 4.6(c)、(d)中,当目标存在平面内旋转或平面外旋转时,覆盖阈值处于 $[0.3, 0.9]$,本章算法的跟踪成功率更高,较传统 KCF 算法,成功率分别提高了 6.0%和 2.3%。



(a) TRE 跟踪精度

(b) TRE 成功率

图 4.5 本章算法与其他跟踪算法整体性能对比

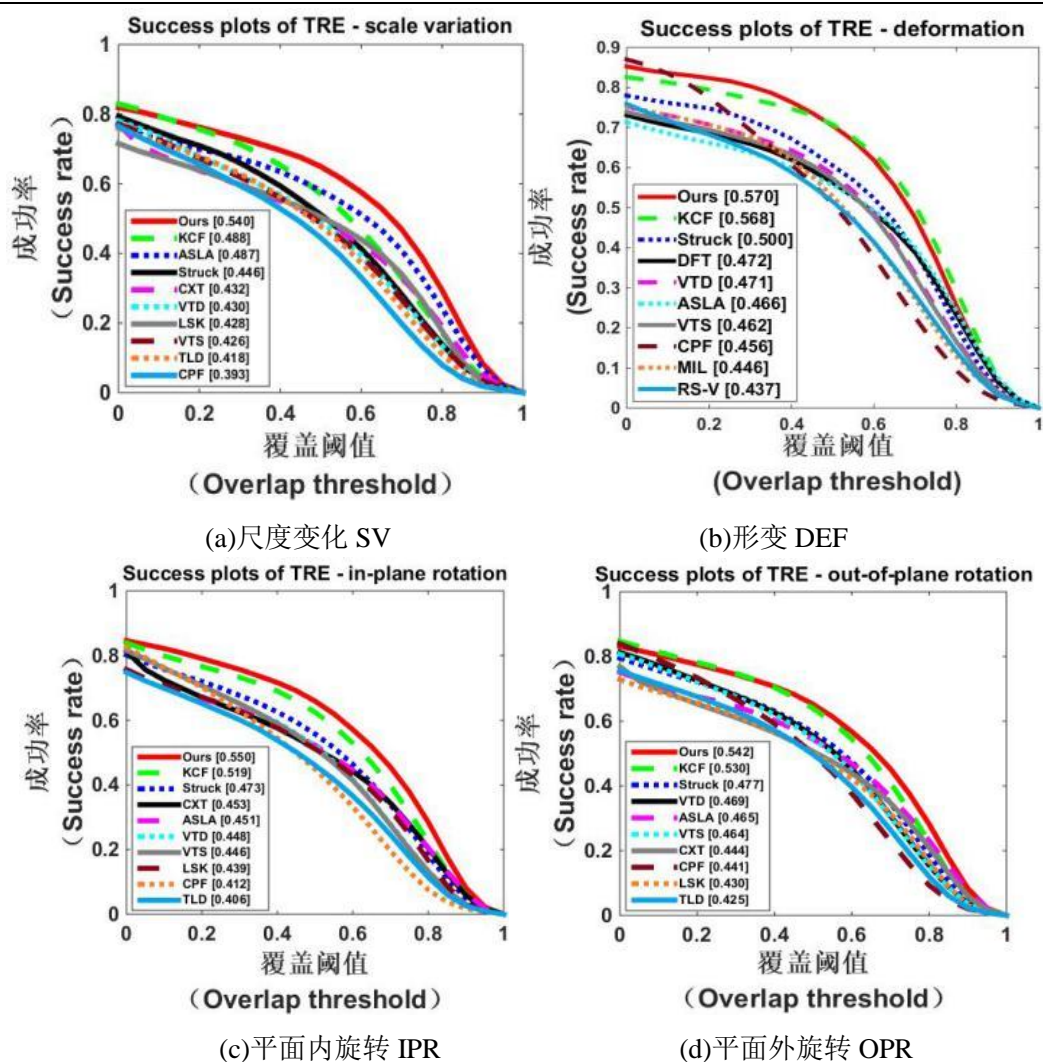


图 4.6 不同属性下算法成功率对比

在图 4.7(a)~(d)中, 当目标存在尺度变化、形变及旋转(内外)时, 误差阈值取 20 像素, 本章算法的精度均高于其他算法。其中, 较排名第二的传统 KCF 算法精度分别提高了 2.0%、5.6%、1.0% 和 4.5%。

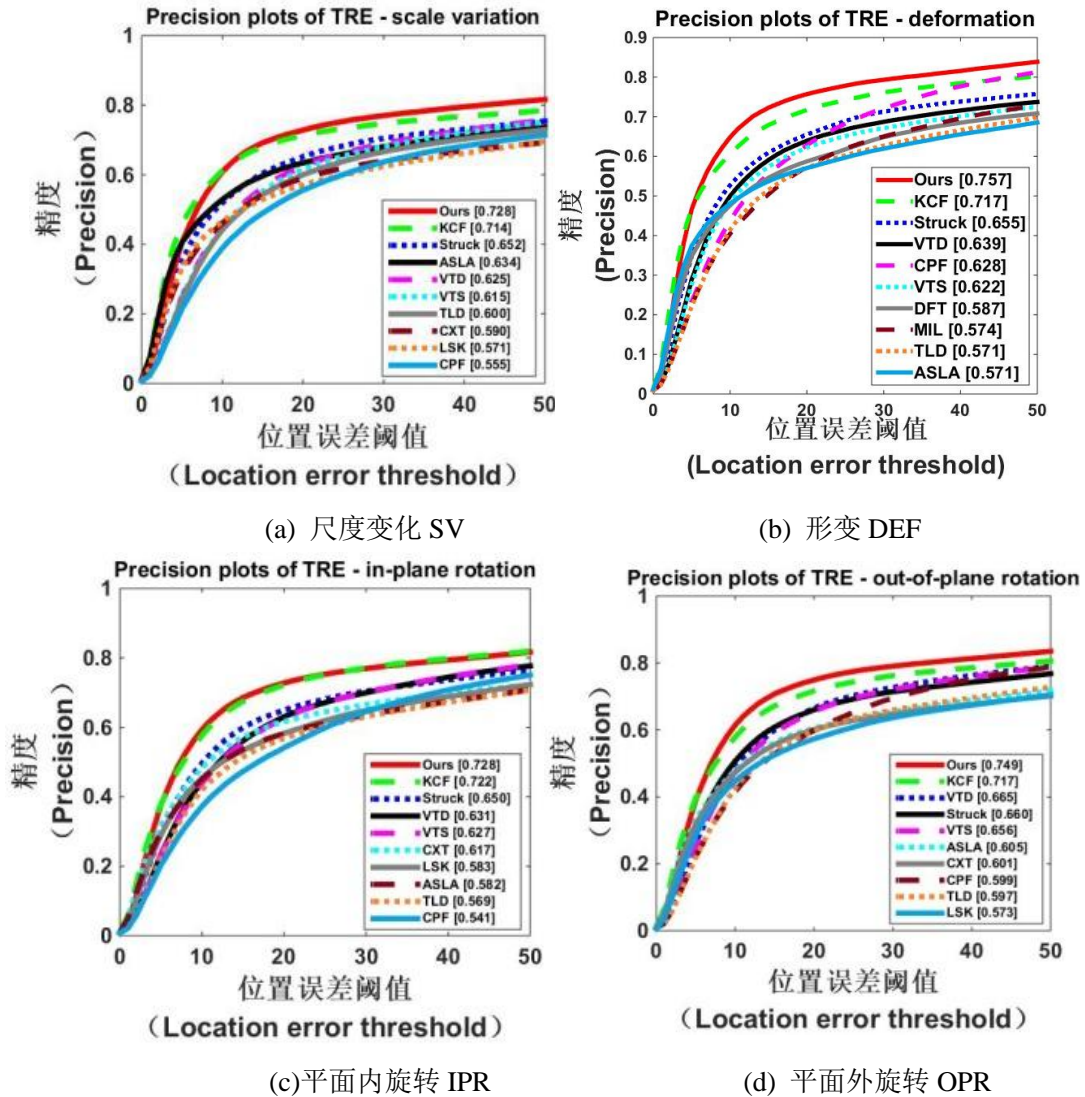


图 4.7 不同属性下的算法精度对比

综上所述，本章算法有效提高了目标存在尺度变化、形变及旋转(内外)等情况时跟踪的成功率和精度，增强了跟踪的鲁棒性，明显改善了跟踪性能。

4.7 本章小结

本章针对目标存在尺度变化及形变情况下 KCF 跟踪效果较差的问题，提出了基于 Radon 变换的尺度自适应 KCF 跟踪算法。在算法中，利用 Radon 变换对噪声不敏感及矩平移伸缩不变性，通过矩特征匹配度峰值确定最优的目标尺度，在减少算法计算量的同时提高了跟踪精度；在利用抛物型学习率曲线的自适应模型更新机制时，根据 Radon 变换的特点，调整了调节参数的取值大小，保证了算法的跟踪精度。通过对 OTB2013 的 50 个视频序列进行测试，结果表明，本文算法在目标存在尺度变化、形变及旋转情况时，具有较好的实时性和较高的跟踪成功率和精度。

第五章 总结与展望

5.1 本文的主要工作

本文基于 YOLO 和相关滤波算法对不同环境下的目标进行检测和跟踪。主要研究内容包括对小目标的检测，以及环境中存在光照变化，目标遮挡、形变和尺度变换时对目标跟踪。论文的主要工作如下：

1. 首先简单阐述了传统 YOLO 算法的网络结构、检测机理和压缩策略，然后，针对 YOLO 算法存在的目标检测准确率低和小目标易漏检等问题，在坐标定位和目标类别判定两个方面提出了损失函数的改进计算方法。前者是基于相对误差的边界框宽高预测损失函数计算方法，该方法即使目标再小，在回归计算中也能得到足够重视，改善了 YOLO 算法对小目标检测率低的不足；后者是基于直觉模糊数的目标类别预测的损失函数计算方法，利用直觉模糊数来代替损失函数中的概率，可对卷积网络得出的特征信息进行全面利用，有效提高了目标检测的准确性。通过选取 COCO 数据集中的车辆图片对改进和传统 YOLO 算法的网络进行训练，并经实验测试，结果表明，本章提出的改进 YOLO 算法在检测目标为小目标或存在遮挡问题时，对比传统的 YOLO 算法，具有较好的实时性和较高的检测精度，检测效果较好。

2. 针对环境中存在遮挡和光照变化时的目标跟踪问题，提出了一种融合运动信息检测的改进 KCF 跟踪算法。首先利用运动信息，通过光流法和帧间差分法预测出目标可能所在的区域，显著缩小了搜索范围，提高算法速度。其次，在传统 KCF 算法的基础上，根据模型更新策略中学习率与响应图峰值的非线性关系，构造抛物型学习率曲线实现对模型的自适应更新，在目标遮挡或光照变化情况下保证了算法的跟踪精度，并通过对相应视频的跟踪实验验证了改进方法的有效性。

3. 针对目标存在尺度变化及形变情况下 KCF 跟踪效果不好的问题，提出了一种基于 Radon 变换的尺度变化 KCF 跟踪算法。在算法中，利用 Radon 变换对噪声不敏感及矩平移伸缩不变性，通过矩特征匹配度峰值确定最优的目标尺度，在减少算法计算量的同时提高了跟踪精度；采用抛物型学习率曲线的自适应模型更新机制，在目标“空洞”或虚假情况下保证了算法的跟踪精度。通过对 OTB2013 的 50 个视频序列进行测试，结果表明，本文算法在目标存在尺度变化、形变、旋转及环境存在光照变化情况时，具有较好的实时性和较高的跟踪成功率和精度。

5.2 后续研究工作展望

本文针对基于 YOLO 和相关滤波的目标检测与跟踪方法进行了一定的研究，但由于时间和经验的不足，仍有一些问题值得进行改进：

1. 当目标存在快速运动及大面积遮挡时，如何提高跟踪目标精度是进一步有待研究的问题；
2. 本文在坐标定位和目标类别判定两个方面提出了损失函数的改进计算方法，提高了小目标检测的精度，降低了漏检率，但当存在大量密集的小目标时，还是会出现漏检的情况。如何解决这一问题，还需要进一步的研究；
3. 实际场景中，视觉的多目标跟踪将会成为未来的热点，如何在保证精度的前提下，提高算法的实时性，对这一问题的研究也是非常有意义的。

参考文献

- [1] 高文, 朱明, 贺柏根,等. 目标跟踪技术综述[J]. 中国光学, 2014, 7(3):365-375.
- [2] Ren C Y, Prisacariu V A, O. Kähler, et al. Real-Time Tracking of Single and Multiple Objects from Depth-Colour Imagery Using 3D Signed Distance Functions[J]. International Journal of Computer Vision, 2017, 124(1):80-95.
- [3] Shastry A C, Schowengerdt R A. Airborne Videos Registration and Traffic-Flow Parameter Estimation[J].IEEE Transactions on Intelligent Transportation Systems ,2005,6(4):391-405.
- [4] Sang J, Guo P, Xiang Z, et al. Vehicle Detection Based on Faster-RCNN[J]. Journal of Chongqing University(English Edition), 2017, 40(7):32-36.
- [5] 乔俊锋, 朱虹, 史静, 等. 一种快速核密度估计背景建模方法[J]. 计算机工程与应用, 2012, 48(5):192-193.
- [6] Abdelwahab M A, Abdelwahab M M. A Novel Algorithm for Vehicle Detection and Tractking in Airborne Videos[C].IEEE International Symposium on Multimedia,2015: 65-68.
- [7] Srinivasan K, Balamurugan P. Object Detection and Movement Tracking in H.264 Compressed Video Based on Optimal Particle Filtering Technique[J]. Journal of Computational and Theoretical Nanoscience, 2016, 13(11):8171-8181.
- [8] Braun M, Rao Q, Wang Y, et al. Pose-RCNN: Joint Object Detection and Pose EstimationUsing 3D Object Proposals[C].IEEE 19th International Conference on Intelligent Transportation Systems, 2016.
- [9] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(6):1137-1149.
- [10] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2016:779-788.
- [11] Wu Y, Lim J, Yang M H. Online Object Tracking: A Benchmark[C].IEEE Conference on Computer Vision and Pattern Recognition, 2013.
- [12] Im J, Jung J, Paik J. Single Camera-Based Depth Estimation and Improved Continuously Adaptive Mean Shift Algorithm for Tracking Occluded Objects[C].Pacific Rim Conference on Multimedia. Springer International Publishing, 2015.
- [13] 尹宏鹏. 基于视觉的目标检测与跟踪研究[J]. 自动化学报, 2016,42(10):1466-1489.

- [14] Bolme D S, Beveridge J R. Visual Object Tracking Using Adaptive Correlation Filters[C].23th IEEE Conference on Computer Vision and Pattern Recognition, 2010.
- [15] João F H, Caseiro R. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels[C]. European conference on Computer Vision - Volume Part IV, 2012.
- [16] Henriques J F. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on PA & MI, 2015,37(3):583-596.
- [17] Weijer J V, Khan F S. An Overview of Color Name Applications in Computer Vision[C]. International Workshop on Computational Color Imaging, 2015: 16-22.
- [18] Zhang K, Zhang L. Fast Visual Tracking via Dense Spatio-Temporal Learning[C]. European Conference on Computer Vision , 2014.
- [19] Danelljan M. Discriminative Scale Space Tracking[J]. IEEE Transactions on PA & MI, 2016, 39(8):1561-1575.
- [20] Li Y. A Scale Adaptive Kernel Correlation Filter Tracker[C]. European Conference on Computer Vision, 2014:254-265.
- [21] Bertinetto L. Staple: Complementary Learners for Real-Time Tracking[C]. International Conference on CVPR, 2016.
- [22] Kutschbach T, Bochinski E, Eiselein V, et al. Sequential Sensor Fusion Combining Probability Hypothesis Density and Kernelized Correlation Filters for Multi-object Tracking in Video Data[C]. IEEE International Conference on AV & SB, 2017.
- [23] Danelljan M. Convolutional Features for Correlation Filter Based Visual Tracking[C]. IEEE on Computer Vision Workshop, 2015.
- [24] Yang H, Zhong D, Liu C, et al. Robust Visual Tracking Based on Deep Neural Networks and Kernelized Correlation Filters[J]. Journal of Electronic Imaging, 2018, 27(2):112-125.
- [25] Yun S, Choi J, Yoo Y, et al. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning[C]. IEEE Conference on CVPR, 2017.
- [26] Yang W, Liu Y, Zhang Q, et al. Comparative Object Similarity Learning-Based Robust Visual Tracking[J]. IEEE Access, 2019, 7:50466-50475.
- [27] Henriques J F. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on PA & MI, 2015, 37(3):583-596.
- [28] 郑秋梅, 王璐璐, 王风华. 基于改进卷积神经网络的交通场景小目标检测[J]. 计算机工程.2019, 45(12): 87-92.
- [29] 邵宇辉. 基于背景建模的微小运动目标检测算法研究[D]. 西安理工大学, 2019.

- [30] 陈正斌, 叶东毅, 朱彩霞, 等. 基于改进 YOLO v3 的目标识别方法[J]. 计算机系统应用, 2020, 29(1): 49-58.
- [31] Joseph R, Santosh D. You Only Look Once Unified, Real Time Object Detection[C]. Proceedings of the IEEE Computer Society Conference on CVPR, 2016:779-788.
- [32] Joseph R, Ali F. YOLO 9000 Better, Faster, Stronger[C]. 30Th IEEE Conference on CVPR, 2017:6517-6525.
- [33] Shafiee M J, Chywl B, Li F, et al. Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video[C]. CVPR, 2017:543-545.
- [34] Redmon J. You Only Look Once: Unified, Real-Time Object Detection[C]. IEEE Conference on CVPR, 2016:779-788.
- [35] He K. Deep Residual Learning for Image Recognition[C]. CVPR, 2015:770-778.
- [36] Hubel D H, Wiesel T N. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex[J]. Journal of Physiology, 1962, 160(1):106-154.
- [37] Fukushima K. Self-organization of A Neural Nnetwork which Gives Position-invariant Response[C]. International Joint Conference on Artificial Intelligence, 1979.
- [38] Fukushima K. Neocognitron: A self-organizing Neural Network Model for A Mechanism of Pattern Recognition Unaffected by Shift in Position[J]. Biological Cybernetics, 1980, 36(4):193-202.
- [39] Atanassov K. Intuitionistic Fuzzy Set[J]. Fuzzy Sets and Systems, 1986, 20: 87-96.
- [40] Xu Z S, Yager R R. Some Geometric Aggregation Operators Based on Intuitionistic Fuzzy Sets[J]. International Journal of General Systems, 2006, 35: 417-433.
- [41] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553):436-444.
- [42] Liu Q, Hu G, Islam M M. Fast Visual Tracking with Robustifying Kernelized Correlation Filters[J]. IEEE Access, 2018, 6:43302-43314.
- [43] 陈志旺, 王航, 刘旺, 等. 抗遮挡与尺度自适应的改进 KCF 跟踪算法研究[J].控制与决策, 2019,34(12):78-85.
- [44] 成科扬, 师文喜, 周博文. 面向复杂场景的鲁棒 KCF 行人跟踪方法[J]. 南京航空航天大学学报, 2019,51(5):625-635.
- [45] 胡屯仓. 适应于光照变化的多区域目标跟踪算法研究[D]. 西安电子科技大学, 2014.
- [46] 杨叶梅. 基于改进光流法的运动目标检测[J].计算机与数字工程, 2019,439(9):108-110.
- [47] 冯丽霞. 对偶空间理论的形成与发展[D]. 西北大学, 2016.

- [48] Berthold K.P. Horn, Brian G. Schunck. Determining optical flow[J]. Artificial Intelligence, 1981, 17(1-3):185-203.
- [49] 吴长侠. 低空对地运动车辆检测与运动特性分析[D]. 中国科学技术大学, 2011.
- [50] Soowoong J, Sangkeun L. Effective Visual Tracking Using Multi-Block and Scale Space Based on Kernelized Correlation Filters[J]. Sensors ,2017, 17(3): 1-17.
- [51] Wu Y. Online Object Tracking: A Benchmark[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013:2411-2418.
- [52] 陈佳兴.基于 KCF 和视觉注意的多特征动态融合视觉目标跟踪研究[D]. 西南交通大学, 2017.
- [53] 李军宏, 潘泉, 陈玉春,等. 基于 Radon 变换的矩不变量及其应用[J]. 西北工业大学学报, 2004,22(3): 392-396.
- [54] 王晷. 图像旋转与尺度变换不变性识别方法研究[D]. 西安电子科技大学, 2008.

致谢

时光似箭，不知不觉间又度过了三个春秋，回想起在南航读研的三年，感悟良多。在这段时光里，伴随着研途瑰丽的风景，感谢各位尊敬的老师和亲爱的同学，正是你们的指导、支持和陪伴让我始终充满勇气，乐观得面对生活，顺利完成学业。

首先要由衷感谢我的导师丁勇副教授，她思想上高屋建瓴，学识广博，对学生关怀备至，细致入微，从学位论文的选题到定稿都倾注了老师无数的心血。无论过去多久，我都会想起老师平日对我真切关心和耐心指导。当我迷茫低落时，她是倾诉解惑的益友，当我懈怠自满时，她是我端正态度的良师。无论未来终会去往何处，我始终会铭记丁勇老师对我的谆谆教诲和无私奉献。

还要感谢姜斌教授，感谢老师为我们提供了良好的相互学习的活动，开拓我们的眼界和思维，感谢同课题组的冒泽慧老师，杨蒲老师等各位老师的帮助，再次向他们表示诚挚的谢意。

特别感谢好友龚文翔，高振龙和王康，感谢师弟卢盼成、何金和汪常建等 305 实验室的同窗一直以来的支持和理解。

特别感谢父母一直以来默默的支持、理解和关爱。家是我坚实的后盾，父母也是我一辈子的榜样。

最后，谨向参加论文评审和答辩的各位老师、专家表达真诚的感谢！

黄鑫城

二零二零年三月

在学期间的研究成果及发表的学术论文

攻读硕士学位期间发表(录用)论文情况

1. 黄鑫城, 丁勇, 卢盼成, 汪常建. 基于 Radon 变换的自适应模型更新 KCF 跟踪算法. 电光与控制. (第一作者, 已录用)
2. 杨勇, 丁勇, 黄鑫城. 改进 APF 与 Bezier 相结合的多无人机协同避碰航路规划. 电光与控制. 2018, 25(11): 36-41. (第三作者, 已发表)
3. 胡忠旺, 丁勇, 杨勇, 黄鑫城. 基于时空关联—网格聚类的多扩展目标跟踪算法. 传感器与微系统. 2019, 38(2): 129-132. (第四作者, 已发表)
4. 卢盼成, 丁勇, 黄鑫城. 基于预判式学习更新策略孪生全卷积网络的目标跟踪. 计算机应用与软件. (第三作者, 已录用)

攻读硕士学位期间发表(录用)专利情况

1. 丁勇, 卢盼成, 黄鑫城. 发明专利: 一种具有预判式学习更新策略的孪生全卷积网络的目标跟踪方法. 申请号: 201910771373.1. (第三作者, 已公开)
2. 丁勇, 杨勇, 黄鑫城. 发明专利: 一种环境信息未知连续状态下基于势函数奖赏 DQN 的无人机路径规划方法. 申请号: 201910437021.2. (第三作者, 已公开)
3. 丁勇, 杨勇, 黄鑫城. 发明专利: 一种改进 APF 与分段 Bezier 相结合的多无人机协同目标追踪方法. 申请号: 201810183193.7. (第三作者, 已公开)
4. 丁勇, 胡忠旺, 黄鑫城. 发明专利: 一种时空近邻目标检测及网格聚类量测划分下的多扩展目标跟踪方法. 申请号: 201710960963.X. (第三作者, 已公开)

参与科研项目

总参通指重点基金项目(TZLDLYYB2014002), 重要参与者.

攻读硕士学位期间获奖情况

2017-2018 年 11 月获得南京航空航天大学硕士研究生学业奖学金“一等奖”.