电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE

论文题目 **基于深度学习的行人跟踪系统**

专业学位类别	工程硕士
学　　号	**201952011904**
作者姓名	罗文斌
指导教师	漆进　　副教授
学　　院	信息与通信工程学院

分类号 _____ 密级 _____

UDC[注1] _____

学 位 论 文

基于深度学习的行人跟踪系统

（题名和副题名）

罗文斌

（作者姓名）

指导教师 漆进 副教授

电子科技大学 成 都

（姓名、职称、单位名称）

申请学位级别 硕士 专业学位类别 工程硕士

专业学位领域 电子与通信工程

提交论文日期 2022 年 3 月 31 日 论文答辩日期 2022 年 5 月 6 日

学位授予单位和日期 电子科技大学 2022 年 6 月

答辩委员会主席 孟凡满

评阅人 陈客松 汪玲

注 1：注明《国际十进分类法 UDC》的类号。

Pedestrian Tracking System Based on Deep Learning

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline	**Master of Engineering**
Student ID	**201952011904**
Author	**Wenbin Luo**
Supervisor	**Associate Professor Jin Qi**
School	**School of Information and Communication Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期： 年 月 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学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、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期： 年 月 日

摘 要

行人跟踪是计算机视觉应用领域中的一项重要任务，是多目标跟踪的主要应用场景之一，有着广阔的应用发展前景。针对以往行人跟踪方法中检测精度不足和频繁遮挡导致跟踪目标身份切换的问题，本文提出了一种基于 Transformer 网络结构的简单高效的无锚多目标跟踪方法。经过一系列详细的实验设计验证了本文提出的方法的可靠性，该方法的性能超过了 FairMOT 基线网络，在 MOT17 测试集上，多目标跟踪指标 MOTA(Multiple Object Tracking Accuracy)达到了 74.9%，IDF1(Identification F1 Score)达到了 72.5%。本文的主要工作如下：

1. 提出基于 Transformer 结构的编解码结构：通过设计基于 Transformer 结构的特征提取网络，并针对提出的骨干网络设计了适合 Transformer 结构的解码网络，有效的提升了检测能力，最终在 MOT17 验证集上，检测指标 AP(Average Precision)上提高了 0.28%，由于目标检测性能的提升跟踪指标 MOTA 和 IDF1 也相应得到提升，并且模型参数量比基线网络少了 0.7M。

2. 提出了交替冻结训练策略和检测框二次匹配策略：针对检测和身份嵌入特征提取两个任务相互竞争导致检测跟踪精度降低的问题，提出了交替冻结检测分支和嵌入特征提取分支的训练策略，经过先冻结嵌入特征提取分支训练 30 轮，再冻结检测分支和编解码网络训练 10 轮，最终模型在 MOT17 验证集上 MOTA 和 IDF1 分别比基线网络提高了 1.7%和 3.3%。基于 FairMOT 匹配跟踪策略，将检测网络生成的检测框分为高低置信度框，首先通过高置信度框与轨迹匹配，然后再使用低置信度框与剩余轨迹进行匹配。在使用了检测框两次匹配策略后，通过交替冻结训练的模型在 MOT17 验证集上 MOTA 和 IDF1 分别提高了 0.9%和 0.7%，并且身份标识切换次数降至 271。

3. 设计开发并部署基于深度学习的实时监控行人跟踪系统：针对行人跟踪任务需求，将该系统设计为三个部分，分别为基于 Android 平台的视频图像采集系统、基于 PyTorch 平台的数据处理系统以及基于 Web 页面的展示系统。系统实现了实时视频的采集和行人的检测跟踪，在视频监控页面展示实时推理视频和当前帧的人数，在数据分析页上展示基于人数的分析图表。

关键词：FairMOT，行人跟踪，Transformer，实时监控系统

ABSTRACT

Pedestrian tracking is an important task in the field of computer vision, and it is one of the main application scenarios of multi-object tracking, which has broad application and development prospects. In order to solve the problem of ID switches of tracking object caused by frequent occlusion and insufficient detection accuracy in previous pedestrian tracking methods, this thesis proposes a simple and efficient multi-object tracking anchor-free method based on Transformer network structure. The reliability of the proposed method is verified by a series of detailed experimental designs. The performance of the proposed method is better than that of FairMOT baseline network. On the MOT17 test set, MOTA(Multiple Object Tracking Accuracy) and IDF1(Identification F1 Score) achieve 74.9% and 72.5% respectively. The main work of this thesis is as follows:

1. Proposed the encoder-decoder structure based on Transformer structure: By designing feature extraction network based on Transformer structure, and designing decoder suitable for Transformer structure according to the proposed backbone network, the detection ability is effectively improved. Finally, the detection index AP(Average Precision) is increased by 0.2% in MOT17 verification set. Due to the improvement of object detection performance, tracking indexes MOTA and IDF1 are also improved correspondingly, and the number of model parameters is 0.7M less than that of baseline network.

2. Proposed the alternating freezing training strategy and the two-time matching strategy of detection boxes: Aiming at the problem that the detection and tracking accuracy were reduced due to the competition between detection and identity embedded feature extraction, a training strategy of alternately freezing detection branch and embedded feature extraction branch was proposed. After 30 rounds of training of freezing embedded feature extraction branch, then 10 rounds of training of freezing detection branch and encoder-decoder network. MOTA and IDF1 of the model on the MOT17 validation set are 1.7% and 3.3% higher than the baseline network, respectively. Based on the FairMOT matching tracking strategy, the detection boxes generated by the detection network is divided into high and low confidence boxes set. The set of high confidence boxes is first used to match the trajectory, and then the low confidence frame

is used to match the remaining trajectory. After using the two-time matching strategy of detection boxes, MOTA and IDF1 of the model trained by alternating freezing on the MOT17 validation set increased by 0.9% and 0.7%, respectively, and ID switches decreased to 271.

3. Designed, developed and deployed a real-time monitoring pedestrian tracking system based on deep learning: According to the requirements of pedestrian tracking task, the system was designed into three parts: video and image acquisition system based on Android platform, data processing system based on PyTorch platform, and Display system based on Web page. The system realizes the real-time video collection and pedestrian detection and tracking, displays the real-time inference video and the number of people in the current frame on the video monitoring page, and displays the analysis chart based on the number of people on the data analysis page.

Keywords: FairMOT, Pedestrian Tracking, Transformer, Real-time Monitoring System

目 录

第一章 绪论

1.1 研究工作的背景与意义

传统的视频监控系统，以视频记录功能为主。随着技术的更新与进步，视频监控也逐渐向着智能化的方向发展，植入了更多对视频监控场景中的目标实例的识别及理解，但现在的视频监控系统功能仍然比较单一，且缺乏稳定性和扩展性。据相关数据统计，超过 70%的视频监控数据和相关研究都是针对于行人进行的。其主要原因在于:大多数视频监控都是固定机位，对于研究运动速度较快和运动范围较广的目标是不友好的，并且行人数据易于采集，应用场景非常的广，如智能安防、体育赛事、智能交通等。

视频下的行人跟踪是多目标跟踪(Multi-Object Tracking, MOT)研究方向中一项关键的子任务，是一项必须持续探讨的科研方向。而多目标跟踪任务是在事前不知道目标数量的情况下,对视频中的行人、汽车、动物等多个目标进行检测,并赋予身份标记(Identity Document, ID)以完成轨迹追踪[1]。不同的目标拥有不同的 ID,以便进行后续的轨迹预测、精准查询等工作。该问题的成功解决可以立即使许多应用受益，如智能视频分析、人机交互、人类活动识别，甚至是社交计算。

在单目标跟踪(Single Object Tracking, SOT)任务中，被追踪的目标的存在通常是能够预先获知的，而在多目标追踪任务中，就必须借助检测步骤来确定出能够自由出入场景的目标[1]。并且同时跟踪多个目标的主要问题来自于多样的遮挡情景、目标与目标之间的相互重叠情景和有时不同的目标也会有相似的外观特征。因此，简单地直接应用单目标跟踪任务模型来解决多目标跟踪任务，通常会导致比较差的结果。因为单目标跟踪模型通常难以区分外观相似的同类但不同实例对象，使得跟踪过程中经常会出现目标漂移和大量的目标身份标识切换错误。

单目标跟踪任务主要研究内容集中在设计复杂的外观模型和运动模型，用来解决目标跟踪中极富挑战性的问题如目标的尺度变化、光照强度的变化、目标的形变和旋转、运动模糊、背景杂波[1]。除了单目标跟踪的问题外，多目标跟踪任务还有额外的两个任务需要解决：确定目标的数量（通常随时间变化）和维持目标各自的身份标识。所以多目标跟踪任务还必须应对更为复杂的几个关键性问题[1]包括:目标被频繁遮挡;目标的跟踪轨迹的初始化和终止;目标类内不同实例间具有相似的外观特征;多目标间的相互重叠的影响。因此，多目标跟踪仍然是目前视频图像处理中一个非常有挑战性的任务，需要视频图像领域的研究人员进行长期的探索。目标跟踪是一个由来已久的方向，但以往的研究大多聚焦在单目标跟踪

任务，而多目标跟踪任务直到最近几年才受到研究人员的密切关注[1]。与图像分类和目标检测等视觉任务相比，多目标跟踪任务主要面临如下几个研究困境：

1. 数据集缺乏且标注困难：由于之前研究大多是集中于单目标跟踪，所以能够使用的多目标跟踪的数据集较少，并且标注多目标场景，如拥挤人群，需要将视频逐帧标记使得标注繁琐。

2. 目标检测结果不精确：由于目前的多目标跟踪框架，大多数都是根据检测器的目标检测结果而实施跟踪，因此检测器的目标检测结果不精确将直接影响到多目标跟踪的跟踪准确度。

3. 频繁的目标遮挡：不同于单目标跟踪，多目标跟踪目标可能被多目标之间干扰遮挡，频繁遮挡会使得运动轨迹碎片化。

4. 目标数量不确定：多目标跟踪中检测出的目标数量不确定，那么在保存特征和运动轨迹预测时将受目标数量影响。

5. 速度较慢，实时性不够：大多数现有方法试图通过两个独立的模型来解决这个问题:首先使用检测模型检测每一帧感兴趣的对象边界框,然后关联模型提取相对应的图像区域每一个边界框的嵌入特征,链接到现有的轨迹或按照一定的指标上重新初始化一个新的跟踪轨迹，但是这样的两个模型使得实时性较差。

计算机视觉建模长期以来一直由卷积神经网络(Convolutional Neural Networks, CNN) [2]主导。从 AlexNet[2]在 ImageNet 图像分类挑战上实现了革命性的性能突破开始，CNN 框架就通过使用更大的模型规模、更加广泛的连接方式和更加复杂的卷积形式而逐渐变得更加强大。CNN 作为各种视觉任务的骨干网络，这些架构上的进步都使得性能提高，广泛提升了整个视觉领域性能。另一方面，自然语言处理(Natural Language Processing, NLP) 中网络架构的演进采取了完全不同于视觉领域的路径，目前最主流的架构是 Transformer[3]。Transformer 是为序列建模和转换任务而设计的一种模型架构，它将注意力机制应用于建模数据中的长期依赖关系[3]。Transformer 在自然语言处理领域所获得的巨大成功促使研究人员们开始思索它的模型结构是否对计算机视觉领域来说也能达到同样的成效，最近研究中发现它在视觉领域的某些任务上也展示了非常大的潜力，如视觉语言联合建模和图像分类和目标检测。

本文将继续探究深度学习相关方法，如卷积神经网络和 Transformer 模型架构，在行人跟踪领域的应用。本文提出的算法模型将紧紧围绕如何避免行人跟踪任务场景中不利因素影响，从而增加检测和跟踪精度，减少误报和漏报的情况，并通过模型的改进和优化以及监控系统的设计实现成功构建了一个视频监控下的行人跟踪系统。

1.2 多目标跟踪的国内外研究历史与现状

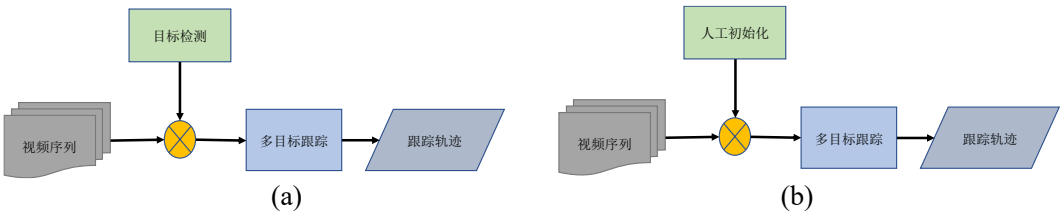


图 1-1 两种主要跟踪方法的流程[1]。(a)基于检测的跟踪(DBT)；(b)无检测跟踪(DFT)

多目标跟踪框架可根据是否依赖于未来视频帧，可以将多目标跟踪分为在线跟踪和批处理跟踪两种方法。在线跟踪方法只能使用以前和当前的视频帧，而批处理跟踪方法可以使用整个视频序列。

在深度学习在视觉领域还未普及时，大多数在线跟踪方法假设目标检测任务是可用的，其工作重点关注数据匹配和轨迹关联，如图 1-1 中(b)所示。例如 SORT[4]跟踪算法以一种非常简单的方式组合位置和运动轨迹。它首先使用卡尔曼滤波[5] 算法来预测轨迹在新帧中的位置，然后计算当前检测框与预测框之间的交并比(Intersection over Union, IoU)值作为相似度，最后通过匈牙利算法[6]进行轨迹匹配跟踪。IOU-Tracker[7]直接使用相邻视频帧的目标检测框的 IoU 值来判断是否是同一个对象。在只关注匹配耗时，而不考虑检测耗时的情况下，IOU-Tracker 跟踪器能达到 10 万 FPS 的推理速度。以上两个方法都因为推理速度快且容易实现而被广泛应用在实践之中。但是由于它们缺乏重复识别能力，使其在具有挑战性的场景中，如拥挤的场景和高速的镜头移动，可能会关联匹配失败。Bae 等人[8]采用线性判别分析（Linear Discriminant Analysis, LDA）提取跟踪目标的重识别（Re-identification，Re-ID）特征，并使用不同帧目标的重识别特征相似度进行匹配，最终取得了比 IoU 匹配更加鲁棒的跟踪结果。Xiang 等人[9] 把在线的多目标跟踪界定为一个马尔可夫决策过程（Markov Decision Process, MDP），首次将在线单目标跟踪和强化学习相结合，来决定跟踪轨迹的生成或消失。批处理方法能够有效地对整个视频序列进行全局优化，使其取得比在线跟踪方法更好的效果。例如，Zhang 等人[10]在多目标跟踪任务中使用了图模型，图模型中的每一个节点都表示了视频帧中的一个检测目标。该算法利用图所特有的结构，使得它比使用线性规划的算法能够更快地找到全局最优解。Berclaz 等人[11]把数据关联匹配视为一种流程优化任务。Milan 等人[12]将多目标跟踪问题界定为能量的最小化问题，并主要探讨了能量函数的建立。其中能量大小取决于在每个帧中每个目标的位置和移动速度及其物理约束。

深度学习的高速发展使得越来越多的研究人员开始使用基于深度学习的目标

检测器，而非基线检测结果，如图 1-1 中（a）所示。例如一些性能较好的方法，如 DeepSORT 2 [13]、POI [14]、CNNMTT [15]、TAP [16]、RAR16wVGG [17]，将目标检测和身份嵌入特征提取视为两个独立的任务。他们先利用标准目标检测模型，如 Faster R-CNN[18]和 YOLOv3[19]，用来定位目标输入图像中任何感兴趣的对象。然后，他们根据模型推理得到目标检测框裁剪输入图像，并将裁剪后的图片注入到一个 Re-ID 网络，用于提取重识别的嵌入特征，随着时间的推移这些特征用于检测框的链接。链接过程通常按照标准目标跟踪关联做法，首先根据检测框的身份嵌入特征和交并比估计代价矩阵，接着再使用卡尔曼滤波和匈牙利算法进行连接任务。在CNNMTT[15]、TAP[16]、RAR16wVGG[17]等少数工作中，研究人员也主张采用更为复杂的匹配关联策略,如组建模和 RNN 建模。两步法的主要优点是能够为每个任务都单独开发出最合适的模板,而不进行其他的折中处理。并且这些方法可以依据预测的检测框去裁剪图像得到图像块，并将图像块调整到相同的尺寸，然后通过 Re-ID 模型获取到重识别特征，这样可以减少比例变化对重识别任务的影响。因此，这些方法[14]在公共数据集上取得了较好的性能。因为目标检测任务和 Re-ID 任务必须单独运行而不能共享模型以便一次性执行，所以它的推理速率往往特别缓慢。因此，两步法的目标跟踪方法很难实现实时推理。在深度学习中，多任务学习[20,21]的研究逐渐成熟，仅需一次性推理的多目标跟踪模型开始受到越来越多的研究。一次性多目标跟踪的核心思想就是在单一网络中同时完成目标检测和嵌入特征信息的获取，以减少推理时间。例如，Track-RCNN[22]在 Mask R-CNN[23]上增加了一个重识别特征提取头部分支，使整个模型能够为每个目标生成边界框和重识别特征。类似地，JDE[24]是基于 YOLOv3[19]模型增加了重识别特征提取分支，FairMOT[25]构建于 CenterNet[26]模型之上并增加了重识别特征提取分支，使得它们实现了近实时的推理速度。然而单步跟踪的精度通常低于两步跟踪的精度。实际上视频的目标检测任务和目标跟踪任务具有一定的关联性，检测任务中可以利用目标跟踪方法来提高具有挑战性帧中的目标检测精度[27,28]。例如，Tang 等人[29]在视频检测中使用了基于相邻帧检测目标管道提高挑战性帧的分类分数。最终小目标的检测率大大提高。但如果视频中有大量对象的时候，这些基于管道的方法非常慢。

在最近的研究中，许多研究人员寻求扩展 Transformer 的应用范围，使其像它对 NLP 和 CNN 对视觉的作用一样可以作为计算机视觉的通用骨干网络。NLP 和计算机视觉任务之间的差异可以用来解释 Transformer 在语言领域的优异表现转移到视觉领域的所面临的挑战。第一个差异主要是数据规模上的差距，Transformer 在自然语言领域应用中基本处理元素为单词标记，但在视觉应用领域，视觉元素

规模上却可能有较大的变化，这也是目标检测等任务中特别关注的问题。由于大多数视觉应用都要适应视觉元素的范围变化，所以现有的大多数基于 Transformer 的固定比例标记的模型都不适合。另一个差异是，图像的分辨率往往比文本段落单词数要高得多。许多视觉任务，例如语义分割，都需要像素级别的高密度预测。这对于应用到高分辨率图像的 Transformer 来说非常的具有挑战性，因为在其自注意力计算的复杂性是原图像大小的二次方。ViT[30]是第一个证明纯 Transformer 在图像分类方面能够获得最先进的性能的著作。ViT 将每个图像视为标记序列，然后将它们提供给多个 Transformer 层进行分类。随后，DeiT[31]进一步探索了数据高效的 ViT 培训策略和精馏方法。最近的方法如 T2T-ViT[32]、CPVT[33]、TNT[34]、CrossViT[35]和 LocalViT[36]对 ViT 进行了定制化的修改，进一步提高了图像分类性能。除了分类之外，PVT[37]是在 Transformer 中引入金字塔结构的第一个作品，它展示了与 CNN 相比，纯 Transformer 主干网在密集预测任务中的潜力。之后，Swin[38]、CvT[39]、CoaT[40]、LeViT[41]、Twins[42]等方法增强了特征的局部连续性，去除固定尺寸的位置嵌入，提高了 Transformer 在密集预测任务中的性能。DETR[43]是第一个使用 Transformer 构建端到端对象检测框架而没有非最大抑制(Non-Maximum Suppression，NMS)的工作。其他研究人员也将 Transformer 用于各种任务，如语义分割、超分辨率、行人重识别、着色、检索和多模态学习。在多目标跟踪任务中，最近也有研究人员提出基于 Transformer 的多目标跟踪框架，目前主要有 TransTrack[44]和 TrackFormer[45]等跟踪模型。然而这些基于 Transformer 的多目标跟踪框架很难部署在实时应用程序中，如何将 Transformer 的能力运用到目标跟踪的任务中去，并且能够部署到实时应用程序中是一个需要不停探索的领域。

1.3 本文的主要贡献与创新

基于深度学习的行人跟踪系统技术使许多应用受益，如智能视频分析、人机交互、人类活动识别，甚至是社交计算，但现在的算法在实际应用时仍有许多不足。针对这些问题，本文对行人跟踪算法网络结构进行改变，最终有效的弥补了之前算法在部分场景的不足，本文的主要创新点和贡献如下：

（1）首次将 Transformer 网络与卷积结构相结合作为检测和嵌入身份特征提取的骨干网络设计跟踪模型，Transformer 在 NLP 任务中表现突出，近期提出的 VIT，SWIN 网络充分说明了 Transformer 结构对于视觉任务同样起作用，并且表现优于当前的 CNN 网络，因此本文基于 Transformer 结构修改特征提取网络，并且基于 Transformer 模型提出混合颈部网络（Mix Neck，MN），本文所提出的跟踪模型对

比已经存在的模型具有更好的性能和结果，有效的提升了检测精度，最终提高了跟踪指标。

（2）重新思考了检测和身份嵌入特征提取的公平性问题，检测任务和身份嵌入特征提取任务是两个完全不同的任务。一般来说，身份嵌入特征提取需要更多的低级特征来区分同一类的不同实例，而检测特征则需要对不同实例进行相似处理。一次性跟踪器中共享的特征会导致特征冲突，从而降低每个任务的性能。大多数的一次性目标跟踪框架都是同时对检测和嵌入特征提取两个任务进行训练，两个任务的相互影响降低了检测和特征提取精度，本文提出交替冻结训练策略，获得了较高的检测精度和跟踪精度。

（3）采用 FairMOT 网络架构，可以同时输出在图像画面中的检测框位置以及检测框内物体的身份嵌入特性，以此加快多目标跟踪的推理速率，然后再利用嵌入特性与边界框 IoU 计算代价矩阵，最后利用卡尔曼滤波和匈牙利算法实现目标的匹配。当目标被遮挡时，目标检测框的置信度就会减小，当小于对目标跟踪的阈值时，就会丢失目标轨迹，从而造成跟踪轨迹的频繁更改，针对以上问题，本文提出一种检测框二次匹配的方法，缓解了目标被遮挡轨迹丢失的情况，进一步提高跟踪精度，减少 Id 切换且具有一定的抗干扰的能力。

（4）基于 Pytorch 框架，使用了 FFmpeg 框架进行推流和拉流,并使用 Nginx 服务器分发视频流,利用 Python 整合以上框架进行调用，最后基于 Spring 和 Mybatis 开源架构[46]实现了基于深度学习的行人跟踪视频监控系统。

1.4 本论文的结构安排

本文的章节结构安排如下：

第一章绪论部分。首先 1.1 节介绍研究工作的背景与意义，其次 1.2 节介绍了研究的课题在国内外的研究现状，调研了国内外相关的研究成果，在 1.3 节介绍本文的主要贡献和创新，最后的 1.4 节说明了论文的整体结构安排。

第二章基于深度学习的行人跟踪的相关理论介绍，主要介绍了本文中所使用到的一些相关基础理论知识，为本文接下来的章节的研究工作奠定理论基础。首先 2.1 节介绍了卷积神经网络结构基础理论，然后在 2.2 节介绍了常用的损失函数，在 2.3 节介绍些常用的优化方法，2.4 节简要介绍经典的 Transformer 网络结构，2.5 节简介与本文研究工作相关的算法模型，2.6 节介绍目标跟踪数据关联的相关知识，最后的 2.7 节介绍了常用的行人数据集。

第三章介绍一种基于 Transformer 结构的实时多目标跟踪算法。第 3.1 节介绍 FairMOT[25]网络作为本文的基线网络，第 3.2 节介绍了一种基于检测框的损失函

数——通用交并比[47]（Generalized Intersection over Union, GIoU）损失，第 3.3 节提出了一种基于 Transformer 的编码器——混合 Transformer 编码器（Mix Transformer, MIT），并为其设计了混合颈部网络（MIX Neck, MN）作为解码器。第 3.4 节对提出的优化方法进行实验验证，并对目标跟踪的相关评价指标进行了说明。

第四章介绍了一种交替冻结训练策略和检测框二次匹配的数据关联策略，并重新审视了身份嵌入特征提取分支对检测网络分支的影响。第 4.1 节重新审视了身份嵌入特征提取任务的平衡问题，提出交替冻结训练策略，第 4.2 节介绍检测框二次匹配策略，第 4.3 节对之前章节提出的方法进行实验验证，并对本文优化后的网络的性能表现进行展示。

第五章对整个行人跟踪系统进行设计与实施，对每个模块进行了详细的设计，并对部署条件和最终调试结果进行展示。第 5.1 节简介了行人跟踪系统的总体框架和运行流程，第 5.2 节介绍了基于安卓平台的推流系统，第 5.3 节介绍了基于 Pytorch 平台的数据处理系统，第 5.4 节介绍了基于 Web[46]页面的展示系统，第 5.5 节对系统的部署要求进行了简介，并对系统进行了联调，并对最终效果进行展示。

第六章全文总结与展望，第 6.1 节对本文的研究内容进行了归纳与总结，第 6.2 节对行人跟踪领域进行进一步分析与展望。

第二章 基于深度学习的行人跟踪理论基础

近年来，基于深度学习的检测方法作为行人跟踪的研究越来越多，本章将对基于深度学习的行人跟踪的相关理论基础进行简单的介绍，并且介绍常用的数据集。

2.1 卷积神经网络

卷积神经网络是具备卷积计算功能的前馈神经网络，其模型结构主要包括三个逻辑层分别为对原始输入进行预处理的输入层、涵盖大部分计算工作的隐含层和根据不同任务输出不同结果的输出层 [48]。在卷积神经网络中最为核心的层次为隐藏层，它包括了卷积层，池化层，激活函数层，全连接层等。

2.1.1 卷积层

卷积层由一组权值共享的滤波核构成，将滤波处理核放在原图像的左上角，将滤波处理核所涵盖的像素区的像素数和相对应的滤波核中数值相乘后，再将乘积结果相加，最终的计算结果被放置在新图像中对应于滤波核中心[48]。图 2-1 中显示了卷积的一步操作，操作完成后内核被移动一个像素，这个过程重复，直到图像中所有可能的位置被过滤。为了解决边缘像素无法覆盖这个问题，卷积网络使用了某种填充方法，较为常见的是零填充的方法，如图 2-1 在卷积图像增加 0 值边界，最终经过卷积操作的结果与输入图像大小保持一致。

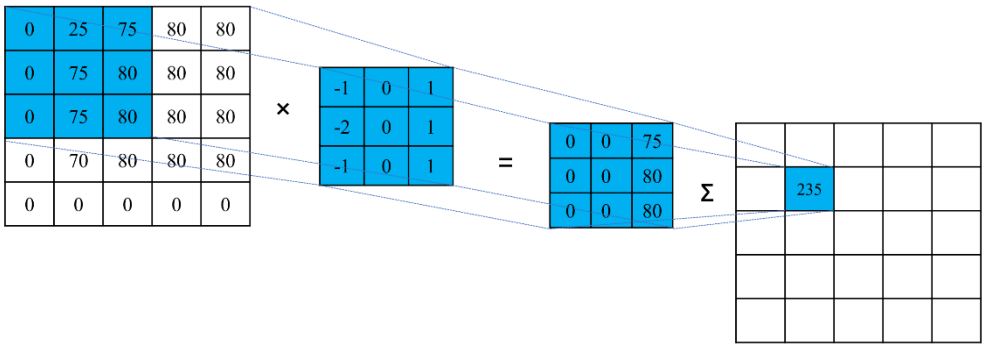


图 2-1 卷积操作

在卷积核中的每一个数字都代表一个权值，权值是输入图像特征与隐藏层之间的连接。在实际运用中，可以通过控制步数大小，填充大小来影响特征层尺寸，

也可以通过控制卷积层滤波器数量控制通道数。卷积核中的权值可以像神经网络一样的方式进行学习，即通过误差反向传播机制进行修正。

2.1.2 池化层

除了卷积本身之外，池化操作[48]也是 CNN 网络的另一个重要组成部分。池化操作通过使用一些函数来总结子区域，例如取平均值或最大值，从而减少特征图的大小。池化层是确保 CNN 网络的后续层能够捕捉到更大规模细节的关键。

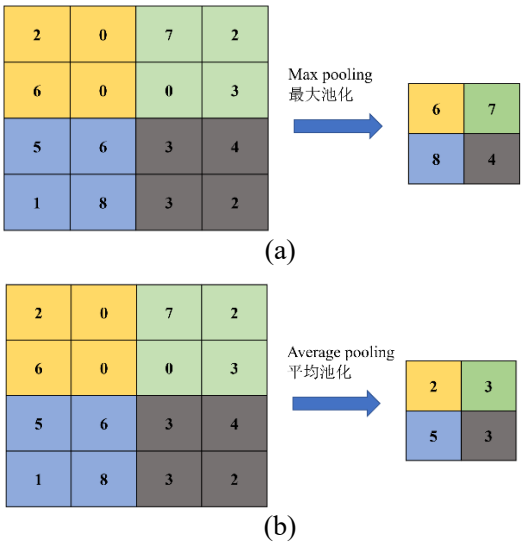


图 2-2 池化操作。(a)最大池化；(b)平均池化

实际上，池化阶段会取一个核，比如 2 ×2 的池化核，并将它传递到整个图像，就像卷积一样。其运行的步数和内核大小基本相等，即一个 2 × 2 池化内核的步数为 2。图 2-2 这个例子将卷积特征图缩放到原图一半大小。经过学习的内核生成的特征图的通道数将保持不变，因为每个通道的特征图会轮流在每个内核上进行池化。因此，特征图经过池化层的计算后将会返回一个与输入特征图相同通道数的数组，但其特征图尺寸将缩小为原来的一半。根据池化核的权值不同又分为最大池化，平均池化等。最大池化以 2×2 的内核为例，如图 2-2（a）所示，取四个点的最大值，这是最常用的池化方法。而平均池化，则为取 4 个点的平均值，如图 2-2（b）所示。

2.1.3 可形变卷积网络

可形变卷积网络[49](Deformable Convolution Network, DCN) 在标准卷积的基础上将二维偏移量加入到网格采样中，如图 2-3 所示，从而使其采样网格能够随意

地变化。偏移量是通过额外的卷积层从先前的特征图中学习得到，所以采集网格变形最终是局部的、密集的或者自适应的，取决于输入特性[49]。

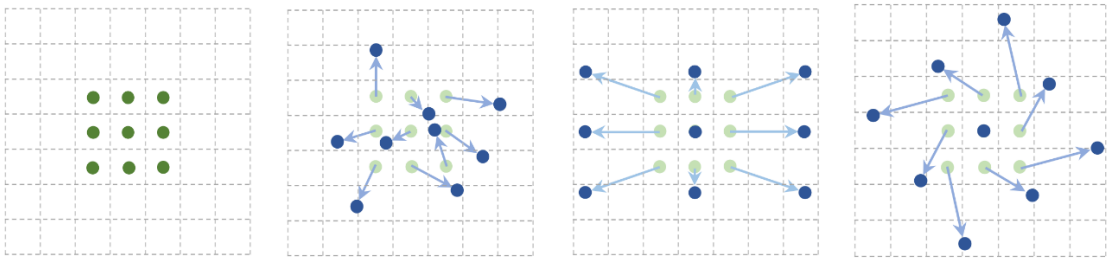


图 2-3 3×3 的标准卷积和可形变卷积的采样位置图[49]。(a)标准卷积采样网格(绿点)；(b)可形变卷积变形采样位置(深蓝点)增强偏移量(浅蓝色箭头)；(c)(d)是(b)的特殊情景，表示可形变卷积涵盖了尺度、纵横比以及旋转等变换

可形变卷积是轻量级的，它只增加了少许参数和计算就达到了学习二维的偏移值的目的。它们可以很容易地替换深度卷积神经网络中的卷积层，并且和标准卷积一样，可以利用反向传播机制进行端到端训练。可形变卷积操作是在标准的卷积操作基础之上加入了一个可学习的参数Δp。以3×3的卷积采样操作为例，标准的卷积采样会对输入x特征图中卷积核覆盖区域进行上采样，而可形变卷积网络会对这些区域进行扩散，使其不再像传统网格形状，这归功于可学习参数Δp，如图 2-4 所示。虽然 DCN 在建模几何变化方面拥有良好的特性，但其对空间的支持却可能大大超出真实的感兴趣的空间范围，导致特征受到无关图像内容的影响。Xizhou Zhu 等人随后提出了第二版的可变形卷积网络[50] (DCNv2)，其提高了可形变卷积的建模能力。主要有两种互补的形式有助于建模能力的增强。第一种形式是扩展使用网络中的可形变卷积层。DCNv2 通过赋予更多的卷积层偏移学习的能力，使其可以在更大范围的特征级别上控制采样。第二种形式是可形变卷积模块中的调制机制，在 DCNv2 网络中的所有样本都会学习偏移量和特征振幅，其中特征振幅将用来调制。因为这两种互补形式，DCNv2 的网络模型同时具备了调整其样本的相对影响和空间分布的功能。

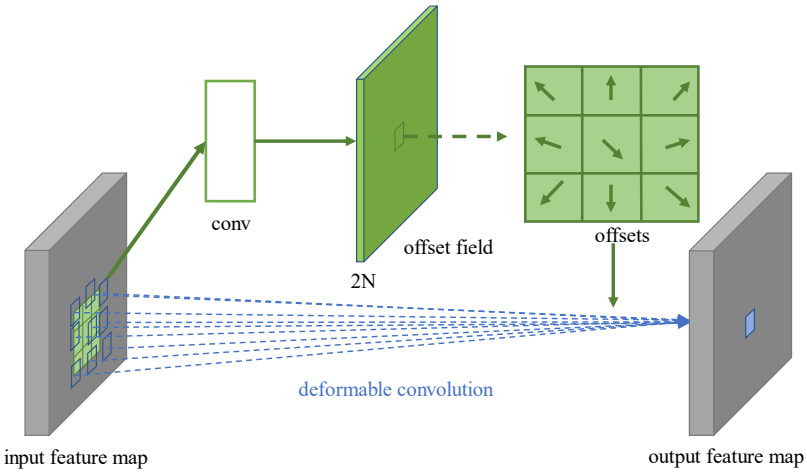


图 2-4 3×3 可形变卷积示意图[47]

2.1.4 全连接层

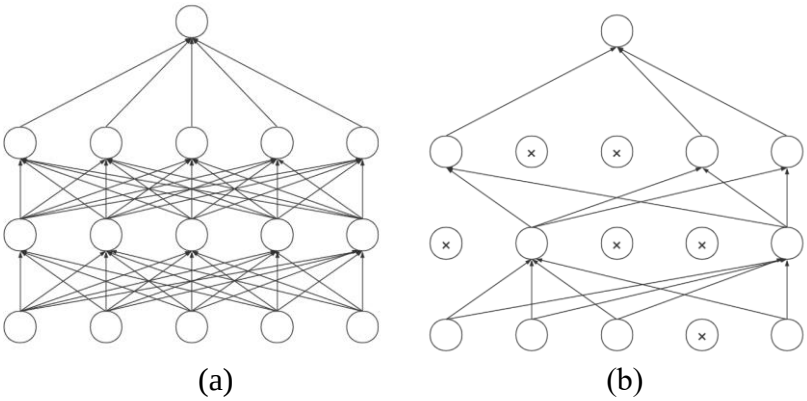


图 2-5 全连接网络[48]。(a)普通全连接网络；(b)带 Dropout 的全连接网络

全连接层[48]，顾名思义就是将上层的每个结点和下层的每个结点都连接起来，其结构如完全二分图，它的主要作用是用来把上层的特征按照一定规则进行综合。全连接的核心运算是矩阵的向量积，如式（2-1）所示。

$$y = W\,x \tag{2-1}$$

其中W为全连接层的权重矩阵。

全连接层的本质就是一个特征空间到另一个特征空间的线性变换。目标空间的每一维都认为会受到源空间的每一维的影响[48]，如图 2-5（a）所示，这个结构和多层感知机（MLP，Multilayer Perceptron）相同，在卷积神经网络中全连接一般出现在最终输出的分类层，而在 Transformer 中经常出现在 Attention 和 MLP 中。为了提高泛化能力，一般全连接层在训练时会增加 dropout 操作，如图 2-5（b）所示。

2.1.5 转置卷积网络

转置卷积[48]又称为反卷积或逆卷积。许多任务想要使用与普通卷积方向相反的变换，这是转置卷积产生的原因。这种变化是从卷积层输出形状特征图到输入形状特征图，并且保持与卷积兼容的连通性模式[48]。例如，卷积网络的解码层可以使用转置卷积网络，当低维特征想要映射投影到更高维空间的时候也可以使用转置卷积网络。同样，卷积网络比全连接网络要复杂得多，在全连接网络通常只需要使用一个转置权值矩阵就可以实现逆向过程。而在卷积网络中，每个卷积都是一次矩阵运算。但是从全连接逆向过程中获得的思想对解决反卷积操作也十分有用。与卷积算法一样，转置卷积网络在轴方向上不会相互作用，所以计算也得到了简化。

转置操作的实现是通过交换卷积的前后通道来实现。使用一个核定义了一个卷积，但它是直接卷积还是转置卷积是由如何计算向前和向后的传递来决定的。例如，虽然核函数w定义了一个卷积，它的向前和向后通过分别与C和C^T相乘来计算，但它也定义了一个转置卷积，它的向前和向后通过分别与C^T和$(C^T)^T$=C相乘来计算。具体的转置卷积操作如图 2-6 所示，一个 3×3 的卷积转置内核以单位步长作用在一个 4×4 的输入上。

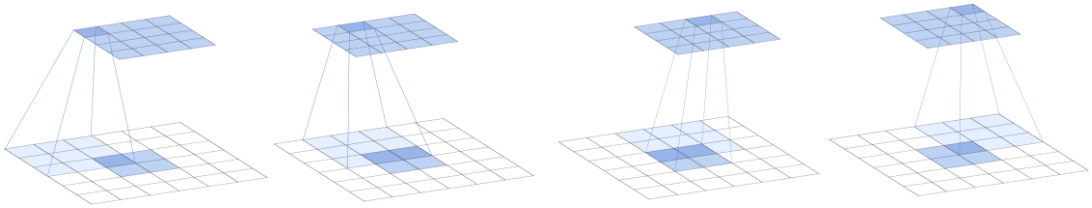


图 2-6 3×3 转置卷积示例[48]

2.1.6 激活函数

在现实中，数据的分布大多是非线性的，但通常神经网络的运算都是线形的。通过引入非线性的激活函数使得神经网络能够胜任非线性任务场景。所以激活函数的最大特点就是非线性的。不同的激活函数根据其特点有不同的应用场景，如将输出限制在(0, 1)和(-1, 1)之间的 Sigmoid 函数和 tanh 函数，由于它对最大最小值进行了限制，使得它更适合做概率值的处理。在更深层网络中则不适宜应用于这二种激活函数，因为他们都会使得梯度消失，取而代之的会用到修正线性单元参数（Rectified Linear Unit, ReLU）以及其变体。

（一）Sigmoid

Sigmoid 函数也称为 Logistic 函数，之所以被成为 Logistic 函数，是因为 Sigmoid 函数可以由逻辑回归（Logistic Regression，LR）中推演出来，同时它也是 LR 模型指定的激活函数[51]。它能够将网络的所有输出映射都在函数的取值范围（0, 1）之间。其公式如式（2-2）, 其图像如图 2-7 所示。

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2-2}$$

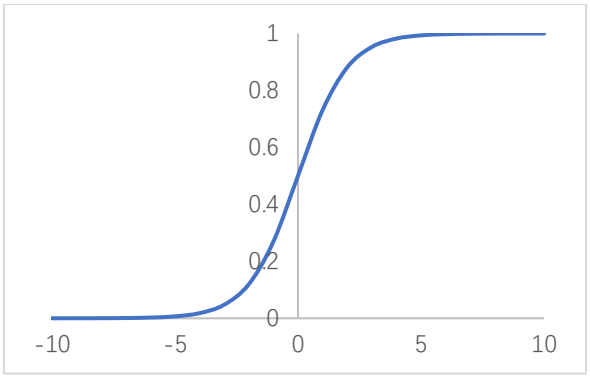


图 2-7 Sigmoid 函数

（二）ReLU

ReLU(Rectified Linear Unit)即修正线性单元函数，ReLU 函数的有效导数是常数 1，这一特性解决了较深的神经网络中出现的梯度弥散问题 [51]，使得深层的神经网络也能够进行训练，ReLU 函数曲线如图 2-8。ReLU 是非线性函数，因为函数的一阶导数值并不是常数。对 ReLU 函数进行求导，当输入值为正的情况下，导数为常数 1，而在输入值为负的情况下，导数为常数 0。ReLU 函数的计算能简单的写成式（2-3）。

$$f(x) = max(0，x) \tag{2-3}$$

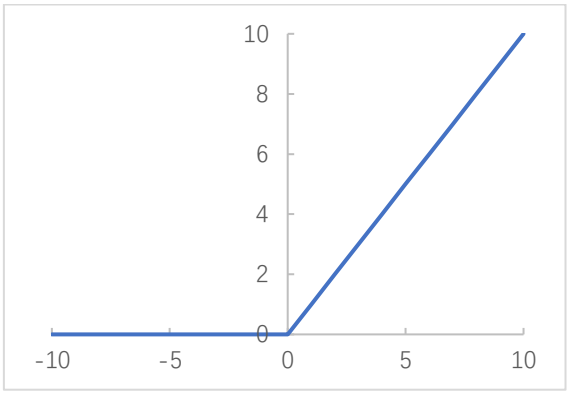


图 2-8 ReLU 函数曲线

但是 ReLU 函数强制的把$x < 0$部分的输出置为 0（置为 0 意味着屏蔽该特征），这样的操作很可能会使得模型无法学习到输入中有效特征。如果学习率设定的过高，有可能会使得网络的大部分神经元陷入“死亡”状态，所以在使用了 ReLU 函数的网络中，不能把学习率设置过高。

（三）Swish

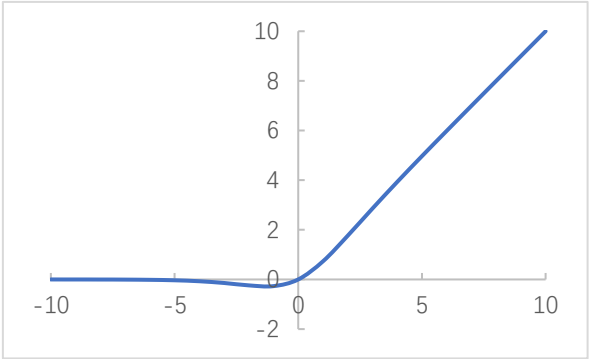


图 2-9 Swish 函数曲线

为了解决 ReLU 函数强制将$x < 0$部分的输出置为 0（置为 0 就是屏蔽该特征），可能会造成模型无法从输入中学习到有效特征的问题，Swish[51]激活函数将 Sigmoid 函数和 ReLU 函数思想相结合，具体公式如式（2-4）。

$$f(x) = x \cdot sigmoid(\beta x) \tag{2-4}$$

其中β是指一个常数或者是一个可以用来训练的参数。Sigmoid 函数如式（2-2），$\beta = 1.0$的 Swish 曲线如图 2-9 所示，导数曲线图如图 2-10 所示。

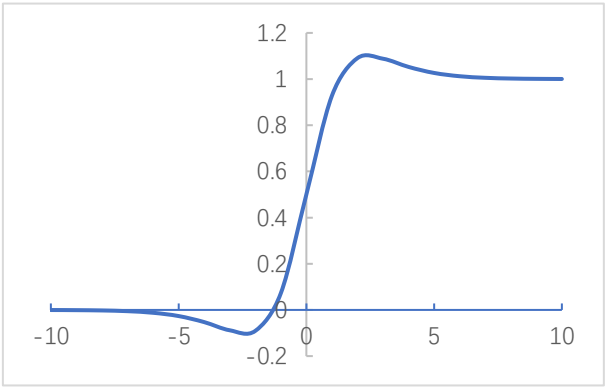


图 2-10 Swish 函数导数曲线

由于 Swish 激活函数具备有下界、平滑、非单调的特性，使得它在更深的神经网络模型上的效果比使用 ReLU 激活函数更好。并且由于β的存在使得 Swish 激活函数成为了介于线性和非线性的平滑函数。

2.2 损失函数

损失函数又称代价函数。在深度学习中，损失函数通常同过输出结果与实际结果进行差异计算，得到模型的当前损失再通过误差反向传递的机制将损失反传给网络模型中去，最终网络模型再修正自己的权重，达到学习的目的。

损失函数按照所属的任务又分为分类问题的损失函数和回归问题的损失函数。回归这一类连续变量预测问题所对应的损失函数常见为最小化绝对误差(Least Absolute Error, L1)，如式（2-5）所示，和最小化平方误差(Least Square Error, L2)，如式（2-6）所示。

$$L(y,\hat{y}) = w(\theta)(\hat{y}-y)^2 \tag{2-5}$$

$$L(y,\ \hat{y}) = w(\theta)|\hat{y}-y| \tag{2-6}$$

式中$w(\theta)$为真实值的权重，y为真实值，$\hat{y}$为模型的输出。

分类这一类离散变量预测问题中常使用的损失函数有交叉熵损失函数（Cross Entropy Loss, CE Loss）和焦点损失函数[52]（Focal Loss, FL）等，交叉熵损失函数如式（2-7）所示。

$$C = -\frac{1}{n}\sum_{x}[y\ ln\hat{y} + (1-y)\ ln(1-\hat{y})] \tag{2-7}$$

其中公式中 x表示样本，y 表示实际标签，$\hat{y}$ 表示预测输出，n表示样本总数。

Focal Loss[52]最初是被设计用于解决目标检测中正负样本不均衡而提出的损失函数，并通过减少对大部分的简单负样本的权重,以均衡对训练过程中不同样本的关注程度。负样本和容易分类的样本过大，导致模型优化的方向偏向于负样本和易于分类样本，所以 Focal Loss 提出设定一个a值以调节正负样本对总损失的共享权重，把a取比较小的数来降低负样本的权重，并通过增加调制系数 γ 来调整容易分类和不易分类的权重[52]。最终公式如式（2-8）（2-9）所示。

$$p_t = \begin{cases} p & if\ y=1 \\ 1-p & otherwise \end{cases} \tag{2-8}$$

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\ log(p_t) \tag{2-9}$$

当目标为正样本时，$y=1$,p为置信分数。Focal Loss 通过a控制正负样本平衡，通过调制参数 γ 控制难易样本平衡。

2.3 优化方法

在网络训练中，通过优化方法让模型在某些条件的约束下，能够决定模型的变量取何值，最终能让目标函数达到最优。模型的优化方法选择要根据训练的模型以及数据而定，选择合适的优化方法能够更快速的收敛，且模型泛化能力更强，在 CV 领域常见的优化算法有：SGD[53]（Stochastic Gradient Descent），Adam[54]，AdamW[55]。

2.4 Transformer 网络结构

在 2017 年，Ashish Vaswani[3]提出了 Transformer 网络结构。它是第一个完全基于注意力机制的序列的转导模型结构，将编码器-解码器体系结构中最常用的递归层替换为多头注意力机制。Transformer 的训练速率比采用循环层或卷积层的架构要快得多，此后 NLP 任务的主流模型都使用的 Transformer 结构。

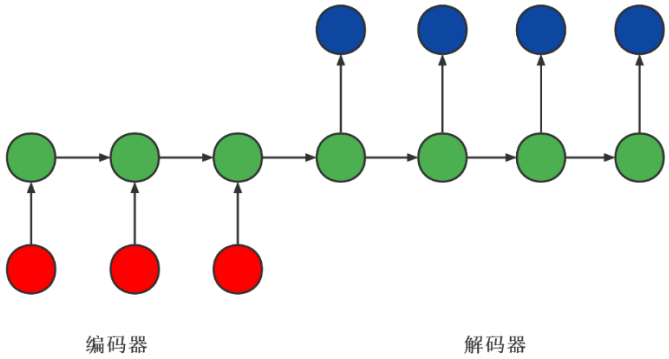


图 2-11 seq2seq 模型结构

与大多数 seq2seq 模型相似都有编码器-解码器结构，如图 2-11 所示。这里的编码器映射输入序列$(x_1, \ldots, x_n)$到一个连续表示的序列$z = (z_1, \ldots, z_n)$。给定z后，解码器生成一个一次生成单个元素字符的输出序列$(y_1, \ldots, y_m)$。在编解码中的每一个步骤中，模型都是自回归的，将先前模型生成的输出作为额外的输入来生成当前的输出。Transformer 模型也遵循这种总体架构，为编码器和解码器使用了堆叠的自注意机制和点对点的完全连接层，分别如图 2-11 的左、右两部分所示。

Transformer 模型基本由五个主要模块所构成，五个模块分别是编码器和解码器，注意力机制，前馈网络以及位置编码。

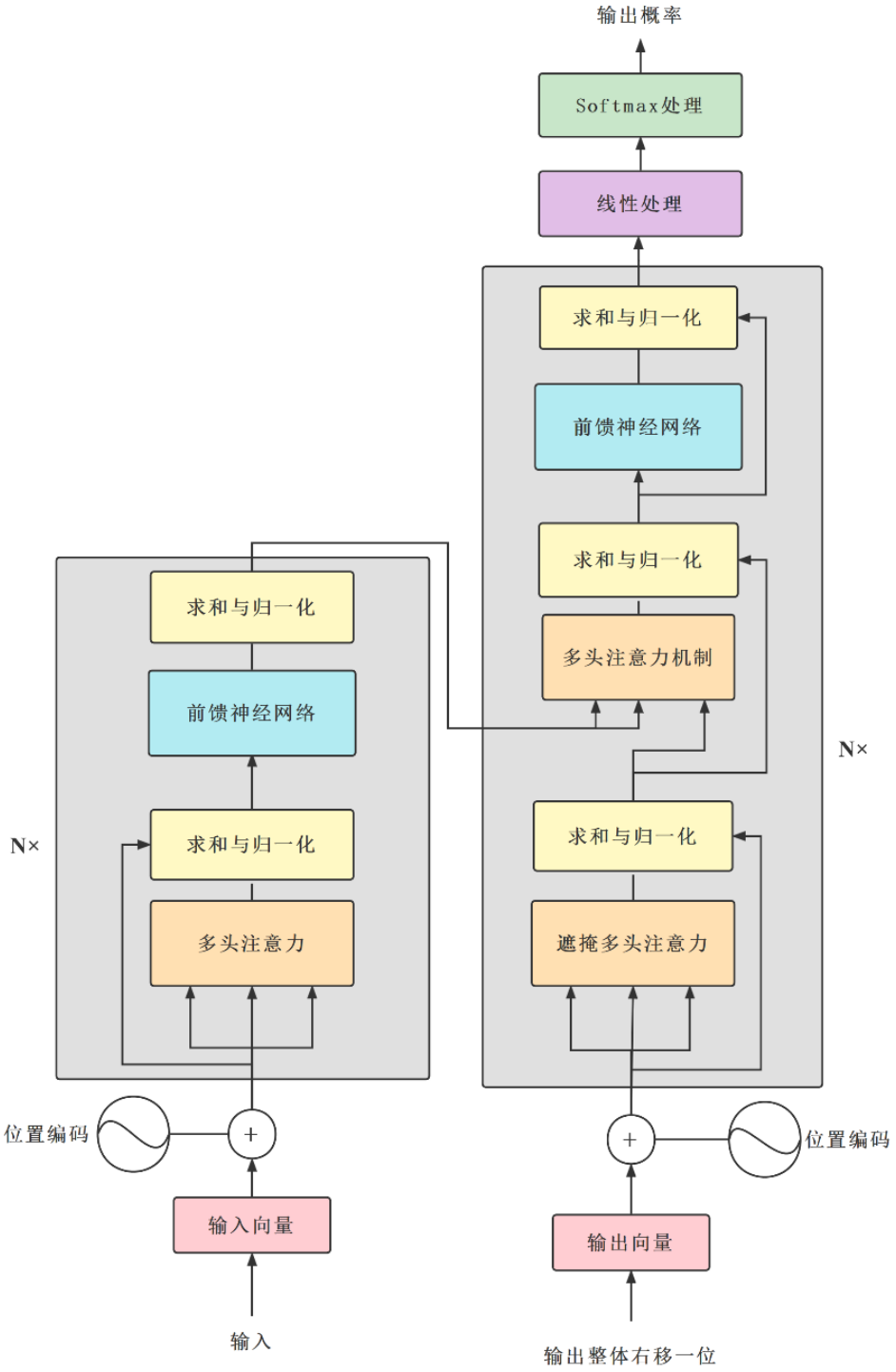


图 2-12 Transformer 模型结构[3]

编码器由 N 个结构相同的层组成，如图 2-12 左侧单元所示。每层都拥有两个子层，其中一个子层为多头的自注意机制，另一个子层为一个简单的、位置上完全连接的前馈神经网络。每个子层的输出可以表示为式（2-10）。

$$Sublayer\_output = LayerNorm\big(x + Sublayer(x)\big) \tag{2-10}$$

其中$Sublayer(x)$是由子层自身完成的函数，LayerNorm 是对每层进行正则化操作。

多头自注意力机制子层是由注意力机制结合而来，而注意力函数可以表述为 Query，Key，Value 的映射，他们利用输入的嵌入向量×乘三个权值不同的矩阵W^Q,W^K,W^V得到。注意力函数输出为 Values 的加权总和，对于每个 Value 的对应权重都是由 Q 和对应的 K 经过一系列的函数计算得到，具体计算流程如图 2-13（a）所示。其计算公式如式（2-11）所示。

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2-11}$$

其中Q,K,V分别是 queries 矩阵，keys 矩阵和 values 矩阵。d_k为Q,K的矩阵维度。除以$\sqrt{d_k}$的目的是因为点乘的结果过大会使得 softmax 函数值位于梯度很小的区域，影响训练。

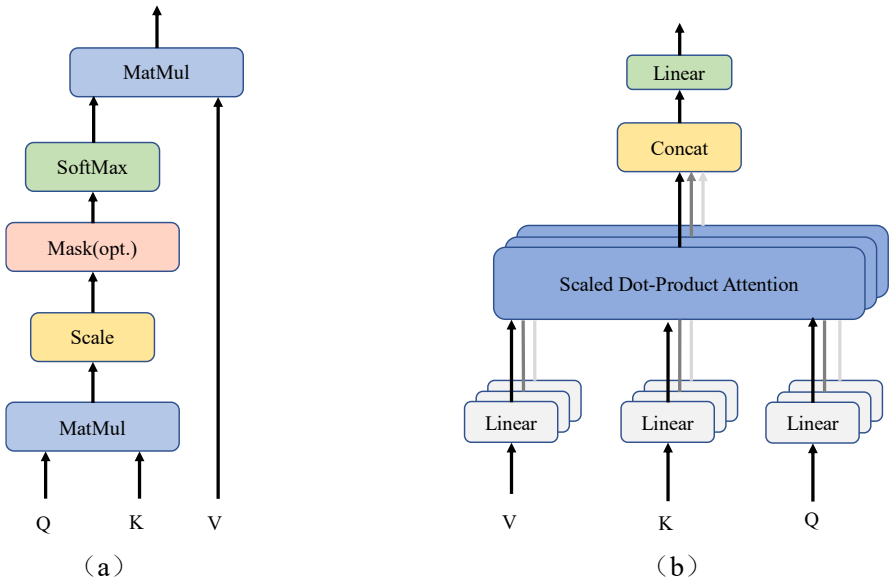


图 2-13 注意力机制[3]。（a）缩放点积注意力机制（Scaled Dot-Product Attention）；（b）多头注意力机制(Multi-Head Attention)

多头注意力机制考虑了各个区域的注意力，使其能够在不同子空间内表现出不同的联结关系，其计算流程如图 2-1（b）所示，公式可以简化为式（2-12）（2-13）。

$$head_i = Attention\left(Q\,W_i^Q,\ K\,W_i^K,\ V\,W_i^V\right) \tag{2-12}$$

其中$W_i^Q \in R^{d_{\text{model}} \times d_k}, W_i^K \in R^{d_{\text{model}} \times d_k}, \in R^{d_{\text{model}} \times d_v}, W_i^O \in R^{d_{\text{model}} \times hd_v}$。

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O \tag{2-13}$$

由于对每个头都进行了降维，即$d_k = d_v = d_{model}/h$，h为头的数量，所以多头注意力机制的最终计算代价与全维的单头注意力机制计算代价相似。除了注意力子层之外，Transformer 的编码器和解码器中的每一层都包含一个完全连接的前馈神经网络，它分别应用于 Q, K, V 的矩阵计算中。并且前馈网络包括两个线性转换，其中一个是 ReLU 激活函数,如式（2-14）所示。

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{2-14}$$

由于模型中不包括递归操作和卷积操作，所以为了使模型能够利用上序列的顺序，需要为模型加入一些关于相对或绝对位置的标志信息。在 Transformer 论文中使用不同频率的正弦函数，如式（2-15）和余弦函数，如式（2-16）。

$$PE_{(pos,2i)} = sin\left(pos/10000^{2i/d_{model}}\right) \tag{2-15}$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{2i/d_{model}}\right) \tag{2-16}$$

其中pos是位置，i是尺寸。也就是说，位置编码的每个维度对应一个正弦信号。波长形成了从2π到$10000 \cdot 2\pi$的几何级数。

2.5 相关算法模型简介

2.5.1 深度聚合网络

深度聚合网络[56]（Deep Layer Aggregation，DLA）使用深层聚合结构来迭代和分层融合特征层次，这种方法使得网络在更少的参数量下获得了更高的准确性。视觉识别需要对从低到高的维度、从小到大的尺寸和由细至粗的清晰度做出更丰富的表现。尽管卷积网络具有深度特征，但是单独一层对于多特征提取是完全不足够的。通过混合和聚合这些特征可以有效的改进模型对“是什么”和“在哪里”的推断。深度聚合网络架构的工作探索了网络骨干网的许多维度，设计更深或更宽的架构。模型结构主要由两个模块所组成，分别是深层迭代聚合（Iterative Deep Aggregation，IDA）和分层深度聚合(Hierarchical Deep Aggregation，HDA)。IDA 模块融合了浅层的底层信息和深层次的语义信息，引入了从浅到深的跳跃连接，如图 2-14 所示。

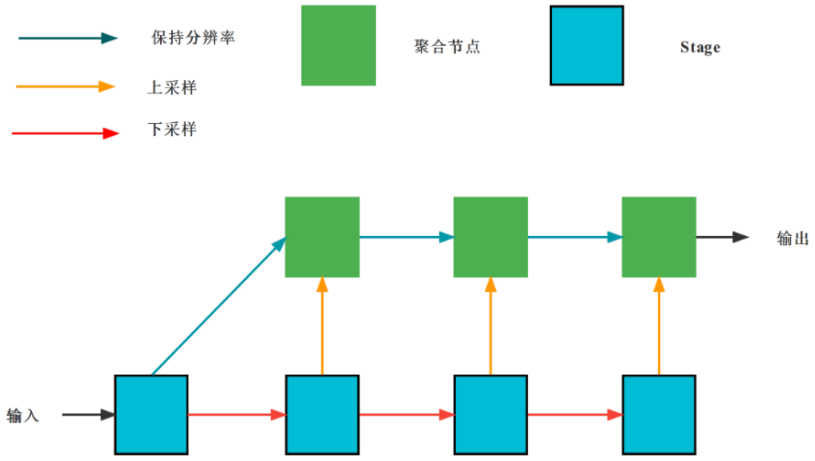


图 2-14 IDA 模块

并且可以用式（2-17）表示网络：

$$I(x_1,\dots,x_n)=\begin{cases} x_1, & if\ n=1 \\ I(N(x_1,x_2),\dots,x_n), & otherwise \end{cases} \tag{2-17}$$

公式中，I表示整个 IDA 模块，N表示聚合模块。$N(x_1,x_2)$表示一个输入为x_1和x_2 的聚合节点，即图 2-14 中绿色方块。

虽然 IDA 模块能够有效的融合各个阶段的特征，但是对于每个阶段内部的多个卷积模块并未进行融合，所以 DLA 在 IDA 模块外提出了 HDA 模块。HDA 模块增加了每个阶段内部的多个卷积模块融合，如图 2-15 所示。

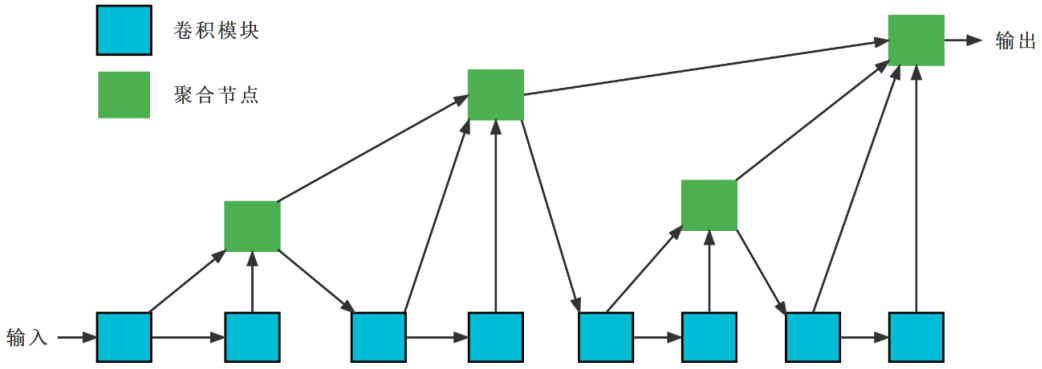


图 2-15 HDA 模块

深层聚合学习可以更好的获取语义信息和空间信息。迭代连接通过将相邻的阶段的特征连接在一起，从而深层的特征和浅层的特征能够更好的融合到一起。并且层次连接通过跨越层级连接也可以更好地传播特征和梯度。最终 DLA 模型的结构如图 2-16。

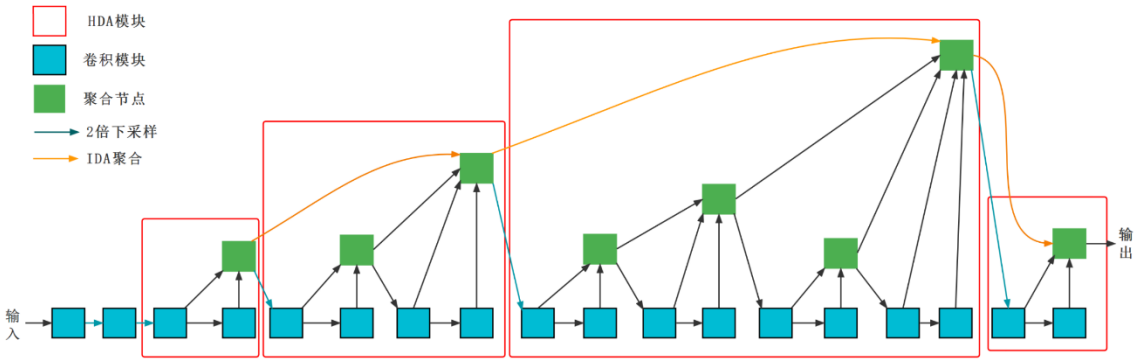


图 2-16 DLA 模型结构图

2.5.2 目标检测算法

目标检测是计算机视觉任务中非常热门的任务，它不同于分类任务只是对每张图片的类别进行区分，还要同时生成对应对象的定位信息。在目标检测任务中常常采用矩形框对目标加以标记,从而确定出该目标的类别，使得图片中每个对象框和对象类别一一对应。

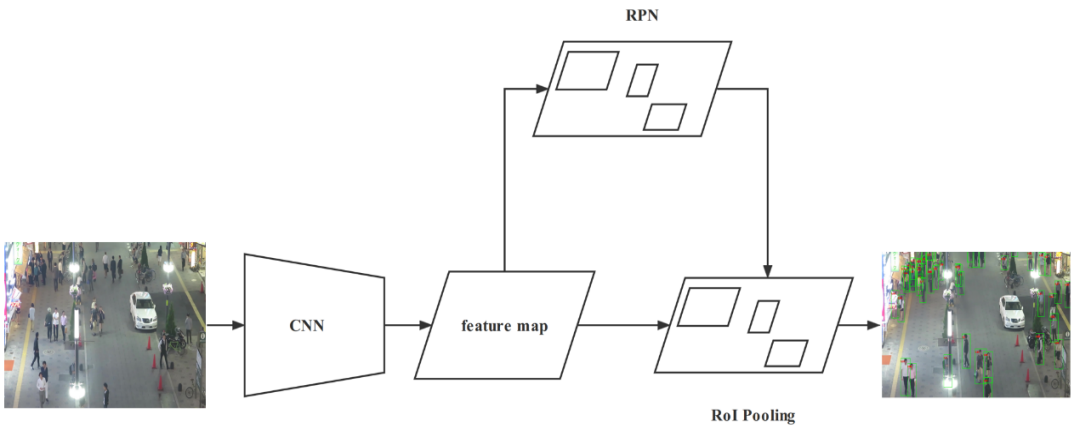


图 2-17 Faster RCNN 结构图[57]

使用深度学习框架的目标检测算法通常被分为二阶段算法和单阶段算法。二阶段的目标检测算法常见的有 RCNN[57]、Fast RCNN[58]、Faster RCNN[18]、Cascade RCNN[59]等。

二阶段目标检测方法，顾名思义将目标检测分成了两个阶段，第一阶段通过骨干网络输出的特征图生成有可能包含目标的预选框（Region Proposal, RP），用于区分前景和背景，第二阶段对一阶段得到的区域建议再次进行边界回归和目标分类。以 Faster RCNN 为例，结构如图 2-17 所示，第一阶段是在一个滑动窗口上产生不同长宽比和尺寸的锚框，取定 IoU 阈值后进行非极大值抑制[60]（Non-

Maximum Suppression，NMS），按真实框标定这些检测框的正负。传入区域建议网络[18]（Region Proposal Net，RPN）的样本数据被整理为目标的边界框坐标和代表是否含有目标的二分类标签。也就是区域建议网络将每个样本都映射为一个代表锚框中是否含有目标的概率值和代表目标边界框的四个坐标位置，最后利用二分类的交叉熵损失和位置坐标回归的损失去训练区域建议网络。第二阶段将区域建议网络得到的候选区域根据概率阈值筛选后，传入 RCNN 的子网络，进行多分类和坐标位置的再次回归。

单阶段模型相较于二阶段模型，减少了推荐区域检出的步骤，直接根据图片预测出结果，目前单阶段目标检测算法常用的有 SSD[61]、YOLO[19]系列、RetinaNet[52]、CenterNet[26]等。单阶段目标检测模型结构以 YOLOV3 为例，如图 2-18 所示。

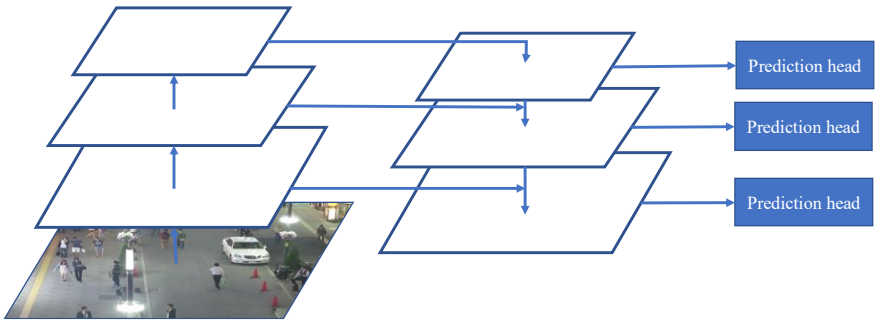


图 2-18 YOLOv3 结构图[19]

单阶段和两阶段模型各有自己的优势，由于单阶段模型不用将生成的锚框再映射到特征图上进行二次分类回归，这使得在速度上单阶段模型比双阶段模型的网络快许多。同样单阶段模型也有其劣势，由于单阶段模型网络学习的锚框有很多，但是只有少量的锚框对最终的学习有利，而大量不利于学习的锚框会影响到整体网络的学习，拉低整体模型的准确率。双阶段模型通过区域候选的机制可以平衡正负样本。

为了解决一阶段网络背景锚框过多导致的正负样本不平衡的问题，CenterNet 网络模型提出了 Focal Loss，将目标置信度损失修改为 Focal Loss。取消了锚框的类别判断，转而使用图像中心点进行预测。

2.5.3 JDE

JDE[24]（Jointly learns the Detector and Embedding model）模型结构在 2020 年 7 月被提出，是一个面向实时的多目标跟踪模型，该方法将目标检测与嵌入特征学习相结合。模型基于一阶段检测器模型，在检测器输出多增加一个分支以输出

学习嵌入特征。然后使用多任务学习的思路设置了损失函数。JDE 网络以 YOLOv3 作为检测模型，预测头结构如图 2-19 所示。

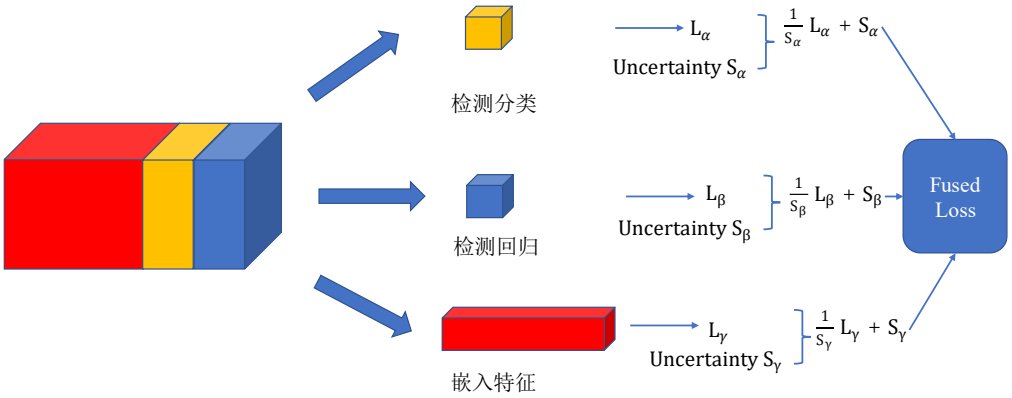


图 2-19 JDE 预测头结构[24]

通过 YOLO 检测器输出检测结果和嵌入特征，随后通过卡尔曼滤波和匈牙利算法进行目标的匹配。JDE 作为一个多目标跟踪架构，允许在一个共享的模型上对目标检测任务和重识别任务进行学习。使得 JDE 模型大大地降低了多目标跟踪架构的运行时间，使其能够以(接近)实时速度运行。

2.6 目标跟踪中数据关联算法

在检测网络将每帧中感兴趣的目标检测出来后，得到了相应的位置坐标和嵌入特征，如何将每帧的数据进行关联是目标追踪所需要关注的问题。常见的数据关联算法主要使用了卡尔曼滤波和匈牙利算法。

（一）卡尔曼滤波

卡尔曼滤波实质上是一个数据融合算法,它通过把所有带有同一检测目的、来源于各种传感器、(可能)具有不同单位大小的数据融合到一起,以达到一个更加精确的目的检测值[5]。卡尔曼滤波的典型使用例子是通过一组有限的、包含噪声的物体位置的观察序列（可能有偏差）预测出物体坐标位置和速度[5]。在目标跟踪中，通过给定目标的初始位置，由预测和测量不断迭代，更新状态向量，最终达到预测的目标位置轨迹的效果。

（二）KM 算法

匈牙利算法(Hungarian Algorithm)是一种组合优化算法，用于解决指派问题，算法的时间复杂度为$O(n^3)$。该算法由 Harold Kuhn 发表于 1955 年，因为该算法基于两位匈牙利数学家的早期研究成果，所以被称作“匈牙利算法”[6]。匈牙利方法是为给定的成本矩阵找到最优分配的算法。因为在匈牙利算法中各个匹配对

象的权重都相等，所以这样得到的是最大匹配结果往往和想要的最佳匹配结果相悖，而 KM(Kuhn-Munkres Algorithm)算法在匈牙利算法的基础上增加对象匹配权重,在多目标跟踪任务中KM算法可以通过统计匹配对象相似程度,从而获得了前后二帧的最大相似度矩阵。KM 算法最终通过求解这个相似度矩阵，从而找到前后二视频帧中真正匹配的目标。

了解匈牙利算法之前，需要了解几个基础概念：

二部图：若图G的顶点可划分为两个非空子集X和Y，使得 G 的任一条边都有一个端点在X中，另一个端点在Y中，则称 G 为二部图记为G（X，Y）。

匹配：设G是一个图，由G中一些不相邻的边组成的集合M称为G的一个匹配。对匹配M中的每条边$e = uv$，其中两端点u和v称为被匹配M所匹配，而u和v都称为是M饱和的。

最大匹配：图G中含边数量最多的匹配称为G的最大匹配。

完美匹配：如果G中的每个点都是M饱和的，则称M是G的完美匹配。

增广路，可扩展路：设M是图G的一个匹配，G的一条M交错路是指其边在M和$E(G) - M$中交替出现的路。如果G的一条M交错路的起点和终点都是M非饱和的，则称器为一条M可扩展路或M增广路。

相等子图: 设G是一个赋权二部图, l是G的可行顶点标号,边(u, v)上的权为$w(uv)$。令$E_l = \{xy \in E(G) \mid l(x) + l(y) = w(xy)\}$，G中$E_l$为边集的生成子图称为$G$的$l$相等子图，记为$G_l$。

匈牙利算法步骤：

1. 任取图G的一个匹配M，设X中M非饱和点的集合为A。

2. 若$A = \emptyset$，则停止，输出当前的M（最大匹配）；否则，任取$x \in A$（一个M非饱和点），记$S = \{x\}, T \coloneqq \emptyset$，转下一步。

3. 若$N(S) \subseteq T$，则不存在从x出发的M的增广路，令$A \coloneqq A - x$，转到第 2 步，否则取$y \in N(S) - T$，转下一步。

4. 若 y 是 M 饱和的, 设 $yz \in M$, 令 $S \coloneqq S \cup \{z\}, T \coloneqq T \cup \{y\}$，转第 3 步，否则获得一条$M$增广路$P(x, y)$，令$M \coloneqq M \oplus E(P), A \coloneqq A - x, y$，转第 2 步。

KM 算法步骤：

1. 给$G =$（X，Y）添加一些顶点和权为 0 的边，使之变成赋权完全二部图，记为G。

2. 从G的任何一可行的顶点标号l开始求出相等子图G_l。

3. 在G_l中执行匈牙利算法，如果求得的G_l的一个完美匹配M，则输出M，算法停止；否则匈牙利算法必将终止预两个集合$S \subset X, T \subset Y$, 且 $N_{G_l}(S) = T$, 此时转到下一步。

4. 令$\alpha_l = \min\{ l(x) + l(y) - w(xy) \mid x \in S, y \in Y - T\}$，对于每个顶点 u，修改其标号如式（2-18）。

$$l'(u) = \begin{cases} l(u) - \alpha_l, \ u \in S \\ l(u) + \alpha_l, \ u \in T \\ l(u), \text{other} \end{cases} \tag{2-18}$$

5. $G_{l'}$代替G_l，转到第 3 步。

修改标号的过程是 KM 算法区别于匈牙利算法的地方。修改的目的是在目前找到的M匹配的基础上增加可行顶点，从而得到增广路。

2.7 数据集

行人跟踪常用的数据集有 MOT17[62]、CrowdHuman[63]、CityPersons[64]、ETH[65]。

MOT17 由 Anton Milan 等人在 2017 年提出的公开的数据集，整个数据集共包含 14 个视频序列，是在 2015 年提出的 MOT15 公开数据集的基础上添加了更为细致的标注和更多的边界框。MOT17 数据集具有更加丰富的图像画面，其中包括不同的拍摄角度、不同的相机运动的情景、更多不同天气状况[63]。MOT17 的部分数据示例如图 2-20。其中 7 个带有标注信息的视频序列图像作为数据集的训练集，另外 7 个视频序列图像用作测试集。主要为 1920×1080 分辨率和 640×480 分辨率的视频组成，其中标注包括 292733 个行人框的标注，并且标注了 ID 信息。该数据集已经成为 MOT 任务最常用的数据集，常使用该数据集来评估多目标跟踪模型的精度。

图 2-20 MOT17 部分数据示例

CrowdHuman 数据集是 MEGVII 在 2018 年提出的基准数据集，用于更好地评估拥挤人群场景中的检测器表现。CrowdHuman 的数据集很大，有丰富的场景及标注，并且每张图都包含了高密度的行人，部分数据示例如图 2-21。CrowdHuman 数据集共包括了 15000 张供模型训练使用的图片、4370 张供模型验证使用的图片以及 5000 张供模型测试使用的图片。整个数据集总共有 47 万人类实例，平均每幅图像有 23 个人类实例，数据集中有大量的不同程度的遮挡场景[63]。训练集中的所有的人类实例都拥有头部边界框、人体可见区域边界框和人体全身边界框的标注。

图 2-21 CrowdHuman 部分数据示例

CityPerson 是 2017 年推出的行人检测数据集与 MOT17 类似，数据多采集于城市的街景，包含了大量的立体场景的视频序列，并且为行人数据提供了大量精细的标注，其中包括检测框和行人身份 ID 的标注。ETH 行人数据则是由 Ess[65]等人在 08 年提出的行人跟踪数据集。

2.8 本章小结

本章主要是为后文的研究内容提供理论依据。首先介绍了卷积神经网络的基础知识后引入 Transformer 架构的相关知识。并介绍了与研究课题相关的深度学习模型、目标追踪模型及目标跟踪常用的数据集。

第三章 基于 Transformer 的实时跟踪方法

本章首先介绍一种将 CenterNet 模型和 JDE 框架相结合的一种行人跟踪算法 FairMOT，然后在该算法的基础上，结合 Transformer 结构，设计一个基于 Transformer 网络结构的实时行人跟踪算法。

3.1 平衡检测与再识别任务的多目标跟踪算法

将多目标跟踪任务定义为在一个网络中进行目标检测和重识别的多任务学习，可以实现两个任务的联合优化，具有较高的计算效率。然而，这两个任务往往相互竞争，需要谨慎处理。特别是，以往的工作通常将身份嵌入特征提取视为次要任务，其准确性受到主要检测任务的严重影响。因此网络偏向于主检测任务，对重识别任务不公平。为了解决这个问题，Yifu Zhang[25]等人提出了一种简单而有效的方法叫做 FairMOT，它基于无锚框的目标检测体系结构 CenterNet 而构建。

在大多数多目标跟踪架构中最要存在三个问题：

（1） 锚框不适合用于提取 ID 特征。锚框最初是为目标检测而设计的。然而锚框不适合提取重识别 ID 特征有两个原因：因为一个锚框可能对应多个目标 ID；多个锚框可能对应一个身份。尤其是在拥挤的场景中，这将导致严重的歧义。

（2） 两个任务之间的特征共享。检测任务和重新标识任务是两个完全不同的任务，需要不同的特性。一般来说，身份嵌入特征需要更低级的特征来区分同一个类的不同实例，而检测特性需要对同类的不同实例相似。一次性跟踪器中的共享特性会导致特性冲突，从而降低每个任务的性能。

（3） 特征维度需求。以往行人重识别方法通常学习高维特征，在重识别任务中取得不错的效果，但在目标跟踪任务中，目标检测任务所需维度远低于身份嵌入特征提取任务，但在维度之间的巨大差异会影响两个任务的执行。更重要的是，对于联合检测和重识别网络来说，学习低维的重识别特征可以获得更高的跟踪精度和效率。这也揭示了多目标跟踪任务和 Re-ID 之间的区别，这是常常被多目标跟踪领域中被忽视的问题。

FairMOT 网络是充分分析了上述问题后提出的一次性跟踪器，具体框架如图 3-1。首先将输入图像送入编码器-解码器网络中，得到 1/4 原图大小的高分辨率特征图，随后将特征图注入到两个并行头部分别用来预测目标的检测框和身份嵌入特征，最后提取预测目标中心处的特征以及检测框进行目标的时序联结。FairMOT 网络采用了无锚的目标检测方法，直接使用高分辨率特征图来估计目标

中心位置，生成相应的检测框，这一操作缓解了锚框的歧义问题，使身份嵌入特征与目标中心能够更好的对齐。

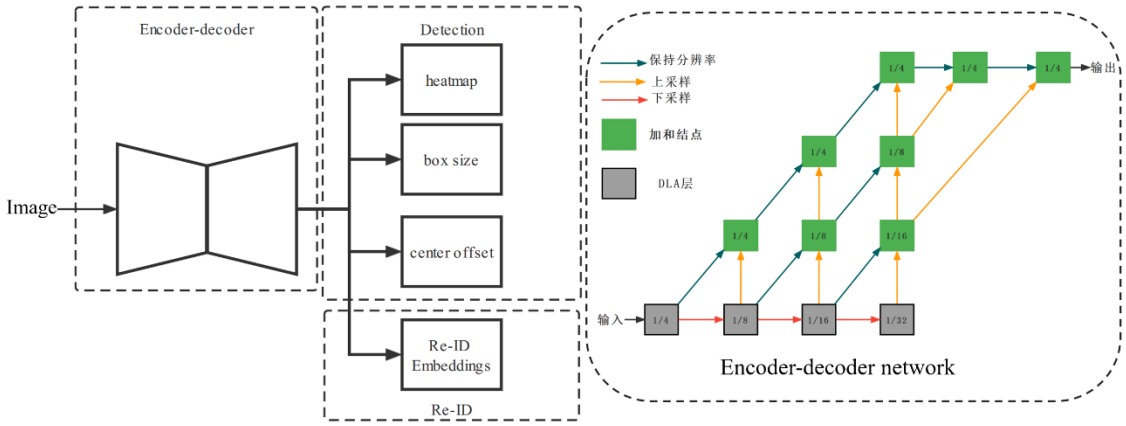


图 3-1 FairMOT 模型结构图[25]

FairMOT 采用 DLA-34 作为骨干。DLA-34 为一种深层聚合网络，拥有大量的低层特征和高层特征之间的跳跃连接，从而在目标尺度变化很大的情况下改善行人检测尺度问题，除了骨干网络外的其他部分与 CenterNet 基本一致。

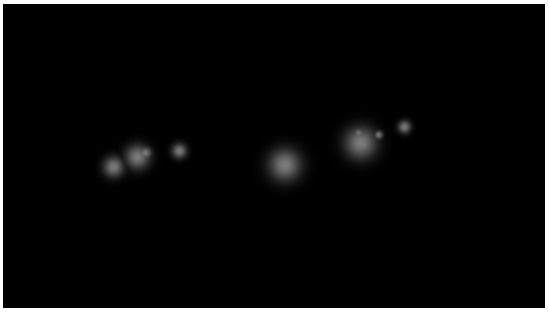
为了根据物体尺度和姿态动态调整接收场，FairMOT 架构将所有上采样模块的卷积操作都改为可行变卷积。这样的修改也有助于缓解对齐问题。如果输入图像的大小为$H_{image} \times W_{image}$，则输出特征图的形状为$C \times H \times W$，其中$H = H_{image}/4$；$W = W_{image}/4$；$C$为检测总类别数，在行人跟踪任务中 C 被设置为 1。检测分支的三个平行头分别用来计算热图、目标中心偏移量和边界框大小。每个头由 3×3 卷积(256 通道)与一个 1×1 的卷积层组成，从而生成最终目标。输出一个大小为$(7 + D) \times H \times W$的密集预测图，其中 D 为身份嵌入特征的维数。将密集预测图分为三个部分(任务):1)尺寸为$2 \times H \times W$的目标中心偏移量;2)尺寸为$4 \times H \times W$的箱体回归系数; 3)尺寸为$D \times H \times W$的身份嵌入特征图；4）尺寸为$1 \times H \times W$的热图。

在生成真实标签时，与普通检测标签不同。假设输入为图片宽度为 W、高度为 H，需要得到关于行人目标的下采样 4 倍的热点图用于训练，这里的热点图生成步骤如下，首先从标签中获得目标的中心坐标$p(x, y)$，然后计算出其下采样四倍后的坐标$\tilde{p}$（$x/4, y/4$）,得到所有目标在图像中的位置后，将其映射到热点图矩阵$Y \in [0,1]^{\frac{W}{4} \times \frac{H}{4}}$中，使用一个二维的高斯核对下采样后的中心进行扩展得到 Y，具体扩展公式如下式（3-1）。

$$Y_{xy} = exp\left(-\frac{(x-\tilde{p}_x)^2 + (y-\tilde{p}_y)^2}{2\sigma_p^2}\right) \tag{3-1}$$

其中σ_p是目标尺度下的自适应的标准方差。如果两个高斯分布相互重叠，取元素方向的最大值。最终生成的热点图如图 3-2 所示。

（a）



（b）

图 3-2 高斯函数生成热点图示例。（a）原图；（b）热点图

热点图的损失函数使用了减少惩罚的适用于逻辑回归的像素级焦点损失函数。其公式如下式（3-2）所示。

$$L_{hm} = \frac{-1}{N}\sum_{xy}\begin{cases}\left(1-\hat{Y}_{xy}\right)^{\alpha} log\left(\hat{Y}_{xy}\right) & \text{if } Y_{xy}=1 \\ \left(1-Y_{xy}\right)^{\beta}\left(\hat{Y}_{xy}\right)^{\alpha} log\left(1-\hat{Y}_{xy}\right) & \text{otherwise}\end{cases} \tag{3-2}$$

其中α和β是 Focal Loss 的超参数，N 为图片中中心点数量。本文中在所有的实验中使用$\alpha = 2$和$\beta = 4$。

因为图像在下采样时的数据都是离散的，使得真实标签的中心点与下采样时得到的标签会形成一定的误差，所以对每个预测目标的中心点位置都增加了局部偏移量的预测，即$\hat{O} \in \mathcal{R}^{\frac{W}{4}\times\frac{H}{4}\times 2}$。用 L1 损失函数来训练偏移量，如式（3-3）。

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - \left(\frac{p}{4} - \tilde{p}\right)\right| \tag{3-3}$$

对应的目标框的大小损失也是使用的 L1 损失，如式（3-4），其中$\hat{S}_{p_k}$为目标框大小预测值，s_k为真实标签值，$\hat{S}_{p_k}$，$s_k \in \mathcal{R}^{\frac{W}{4}\times\frac{H}{4}\times 2}$。

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{p_k} - s_k\right| \tag{3-4}$$

在损失函数方面，FairMOT 也是考虑到不同任务之间权重关系，为了更好的学习到身份嵌入特征，使用了使用 Kendalld 等人[66]在 18 年提出的不确定性损失来自动平衡检测和重新识别任务，如式（3-5）（3-6）所示。

$$L_{detection} = \gamma_1 L_{hm} + \gamma_2 L_{off} + \gamma_3 L_{size} \tag{3-5}$$

$$L_{total} = \frac{1}{2}\left(\frac{1}{e^{w_1}}L_{detection} + \frac{1}{e^{w_2}}L_{identity} + w_1 + w_2\right) \tag{3-6}$$

γ_i为检测内部损失权重。L_{identity}是目标的身份嵌入特征损失，将对象 ID 嵌入看作分类任务，相同 ID 的视为一类，最终使用交叉熵损失函数。其中 w1 和 w2 是平衡这两个任务的可学参数。这些都是比较估计的方法，用来获得损失训练整个网络。

后续的跟踪部分与 JDE 网络基本一致，利用边界框交并比、身份嵌入特征和卡尔曼滤波来计算检测框之间的相似度。并在此基础上，采用了匈牙利算法来处理分配问题。如果只使用检测框的 IoU，会造成大量的 ID 切换错误。这对于拥挤的场景和快速的摄像机运动来说尤为明显。在仅使用重识别特征就能显著增加了 IDF1，并减少了身份标识切换的数量。另外，加入了卡尔曼滤波可以得到更平滑的轨迹，以便于进一步降低 ID 切换的数量。当一个对象被部分遮挡时，它的身份嵌入特征变得不可靠。在这些情形下,利用边界框的 IoU、身份嵌入特征以及卡尔曼滤波，能够达到更优异的跟踪性能。

3.2 GIoU 损失函数

本文在复现了 FairMOT 方法的时候发现了模型训练时对目标框尺寸回归使用的损失函数为 L1 Loss，使用 L1 loss 会导致检测框的回归精确度较低，并且模型推理出的边界框也不能很好的贴合到目标本身。因此本小节对基线网络所使用的损失函数进行改进，以提高检测精度，进而提高跟踪算法的跟踪精度。

边界框 IoU 是目前目标检测用以评估检测精度的评估指标之一，其计算公式如式（3-7）。

$$IOU = \frac{|A \cap B|}{A \cup B} \tag{3-7}$$

其中任一两个凸边形$A, B \subseteq S \in R^n$。$|A \cap B|$代表两个框的交集面积，$A \cup B$代表两个框的并集面积，如图 3-3（a）绿色代表真实标签，蓝色为预测值，（b）中橘色代表了两个框的交集，（c）中紫色代表了两个框的并集。

（a）　　（b）　　（c）

图 3-3 交并集示意图。（a）原图；（b）交集；（c）并集

对于轴对齐的二维的边界框，使用 IoU 损失能够直接用于目标坐标的回归损失，但是 IoU 损失却没有办法作用于完全无重叠的边界框。GIoU 弥补了 IoU 不能计算不重叠案例，并且遵循如下三个原则：

a) 使得 GIoU 与 IoU 定义一样，将对象的形状属性作为感兴趣的区域属性用于比较；

b) 保持了 IoU 的尺度不变性；

c) 当比较目标的边界有重叠的情况下，保证了 GIoU 与 IoU 具有强相关性。

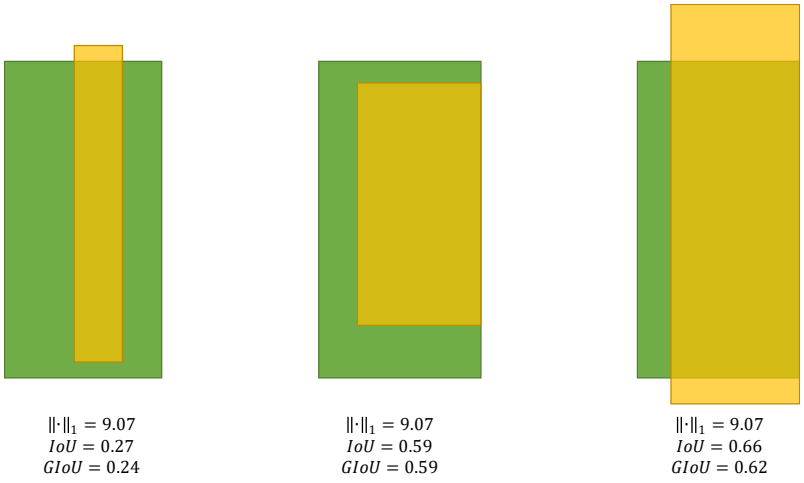


图 3-4 GIoU 与 L1 对比图[47]

当模型使用 L1 Loss 作为检测框尺寸的损失函数的时候会出现如图 3-4 所示的一些问题，图中检测框由中心坐标和检测框长宽(x_c, y_c, w, h)表示。对三个样例使用$l_1 - norm$距离，在图中可以看到两个矩形之间的l_1数值是完全相同的，但是它们的 IoU 值和 GIoU 的值差异非常的大，所以优化检测框的坐标的距离损失并不意味着最大化 IoU 指标。从图 3-4 可以清晰的看到，对于图中的目标来说，$l_1 - norm$距离的局部最优值不一定就是 IoU 的局部最优值[47]。另外，与 IoU 相反，基于上述参数表示定义的 n-范数目标会随着问题的具体规模而变化。因此，几对具有相同重叠程度但由于视角等因素而具有不同尺度的边界框将产生不同的损失值。例如，在中心坐标和尺寸表示中，(x_c, y_c)定义在位置空间上，而(w, h)属于尺寸空间[47]。随着更多参数的加入，例如旋转或添加更多维度到问题中，那么计算复杂度会不断增加。

表 3-1 GIoU 算法

输入：任意两个凸形A,B $\subseteq S \in R^n$
输出：GIoU
1　为 A，B 寻找一个最小外接凸形目标 C$\subseteq S \in R^n$
2　$IOU = \frac{\lvert A \cap B \rvert}{A \cup B}$
3　$GIoU = IoU - \frac{\lvert C \backslash (A \cup B) \rvert}{\lvert C \rvert}$

GIoU 计算方法如表 3-1 所示，对于两个任意的凸形A,B $\subseteq S \in R^n$，算法首先找到最小的凸形 C$\subseteq S \in R^n$使其可以同时包围 A 和 B。为了比较两种特定类型的几何形状，C 可以来自同一种类型。

(a)　　(b)

图 3-5 GIoU 中差集示意图。(a)原图；(b)差集

例如图 3-5 中两个矩形框 A，B 分别对应绿色的真实标签，蓝色预测标签，红色 C 是包围它们的最小的矩形框。接着，使用算法计算 C 所占的面积与 A，B 并集的面积之间的差值，如图 3-5（b）中黄色区域，然后除以 C 所占的总面积，使 C 成为一个标准化的度量，最后再使用 IoU 值减去这个比值，就能得到 GIoU 的表示值。

由于行人跟踪是基于检测结果进行的，所以检测精度也会影响跟踪精度，FairMOT 的检测框尺寸大小使用了 L1 损失函数，使得最终的框与真实标签框有不少差距，因此本文将 FairMOT 方法的目标框尺寸大小损失函数修改为 GIoU 损失函数，修改后的检测损失如式（3-8）（3-9），从本章的实验测试结果表 3-1 来看，GIoU 的加入确实增加了检测精度，相应的跟踪精度也提高了。

$$L_{giou} = 1 - GIoU \tag{3-8}$$

$$L_{\text{detection}} = \gamma_1 L_{\text{hm}} + \gamma_2 L_{off} + \gamma_3 L_{giou} \tag{3-9}$$

3.3 基于 Transformer 的行人跟踪算法

Transformer 编码器在图像领域也能取得很好的成绩，已经被很多优秀工作所证实，本小节试图基于 Transformer 网络结构设计全新的编解码结构运用到 FairMOT 基线网络中去，使整体网络的检测精度得到提升。

3.3.1 混合 Transformer 编码器

Transformer 结构在视觉领域已经取得耀眼的成绩，在很多视觉任务分支上都超过了卷积神经网络，本文也将使用 Transformer 骨干网络运用到目标跟踪网络中，用以解决目标跟踪中检测能力不足的问题。

本节提出的一种混合 Transformer 编码器（Mix Transformer, MIT），其部分设计的部分灵感来自于金字塔视觉卷积网络[37]（Pyramid Vision Transformer, PVT），混合了卷积神经网络，具体设计框架如图 3-6 所示。MIT 编码器可以生成高分辨率粗特征和低分辨率精细特征。给定一个大小为$H \times W \times 3$的图像，首先将其分成大小为4×4的补丁。与使用大小为16×16的补丁的 ViT 模型相反，使用较小的补丁更有利于密集的预测任务。然后使用这些补丁作为输入，在 MIT 编码器中获得$1/4,1/8,1/16,1/32$的原始图像分辨率大小的多层特征。

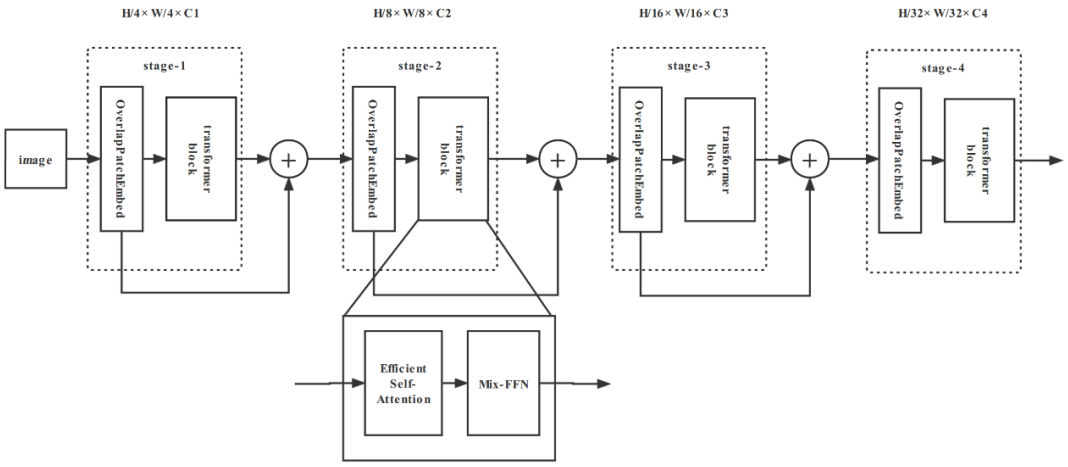


图 3-6 MIT 网络结构

MIT 编码器主要针对 VIT 做了如下 5 点的改进。

1. 新增分层特征表示：与只能生成单分辨率特征图的 ViT 不同，该模块的目标是，给定一个输入图像，生成类似 CNN 的多级特征。这些特征提供了高分辨率的粗特征和低分辨率的细粒度特征，通常可以提高目标检测的性能。本文通过在每个 Transformer 层级模块前增加 Patch merge 层来进行降分辨率。这样本文的网络可以和其他 CNN 网络一样得到 1/4，1/8，1/16，1/32 的分辨率特征图。

2. 新增重叠的补丁合并：给定一个图像补丁集合，ViT[30]中使用的补丁合并过程是将$N \times N \times 3$补丁块统一为一个$1 \times 1 \times C$的向量。这可以很容易地扩展，将一个$2 \times 2 \times C_i$特征路径统一为一个$1 \times 1 \times C_{i+1}$的向量，以获得分层特征映射。使用这个方法可以将层次特征从原来的特征图 F_1 $(H/4 \times W/4 \times C_1)$缩小到 $F_2(H/8 \times W/8 \times C2)$，然后迭代层次结构中的任何特征映射。这个过程最初被设计用来组合不重叠的图像或特征块。因此，它不能保持这些特征块周围的局部连续性。相反，本文使用重叠的补丁合并过程。为此，本文重新定义 K、S 和 P，其中 K 为补丁块大小，S 为相邻两个补丁块之间的跨度，P 为填充大小。设置 K = 7, S = 4, P = 3, K = 3, S = 2, P = 1 进行重叠补丁合并，得到与不重叠过程相同大小的特征。

3. 使用高效 Self-Attention：编码器的主要计算瓶颈是自注意层。在原始的多头自我注意过程中，每个头的 Q、K、V 具有相同的维度$N \times C$，其中$N = H \times W$为序列长度，自注意模块计算如式（2-11）所示，而这一过程的计算复杂度为 $O(N^2)$，这对于大分辨率的图像来说是不允许的。相反，本文使用 PVT[37]中

引入的序列约简过程。该过程使用约简比 R 对序列的长度进行约简，如下式（3-10）（3-11）所示。

$$\widehat{K} = Reshape\left(\frac{N}{R}, C \cdot R\right)(K) \tag{3-10}$$

$$K = Linear(C \cdot R, C)\left(\widehat{K}\right) \tag{3-11}$$

式中 K 序列被减少，线性层采取C_{in}维度的张量作为输入,并生成一个C_{out}维度的张量作为输出。因此，新 K 具有$(N/R) \times C$ 维，使自注意机制的复杂性由 O(N^2)降低到 O(N^2/R)。

4. 使用混合前馈网络：ViT 模型中使用位置编码来引入位置信息。但是位置编码的分辨率是固定的。因此，当测试分辨率与训练分辨率不同时，需要对位置码进行插值，这往往会导致精度下降。为了解决这个问题，CPVT[33]使用3×3卷积网络来实现数据驱动位置编码。位置编码实际上对于目标检测是不必要的。相反，可以直接将3×3卷积网络引入到前馈网络中，并考虑了零填充对泄漏位置信息的影响，如图 3-7 所示。混合前馈网络可表示为式（3-12）。

$$\boldsymbol{x}_{out} = MLP\left(GELU\left(Conv_{3\times3}\left(MLP(\boldsymbol{x}_{in})\right)\right)\right) + \boldsymbol{x}_{in} \tag{3-12}$$

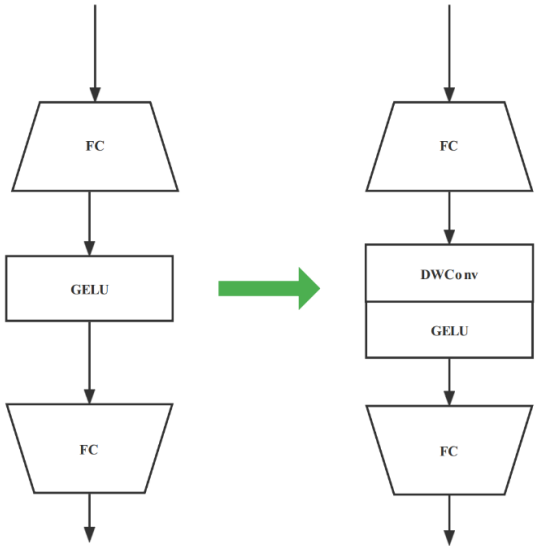


图 3-7 混合前馈编码

5. 跳跃连接：重叠补丁合并后的特征与 Transformer 模块后的特征相加和，具体结构如图 3-8 所示，利用可学习参数w对其进行调节。使模型比 PVT 模型拥有更多的低层和高层特征之间的跳跃联系，使模型能够更适合目标检测的任务。

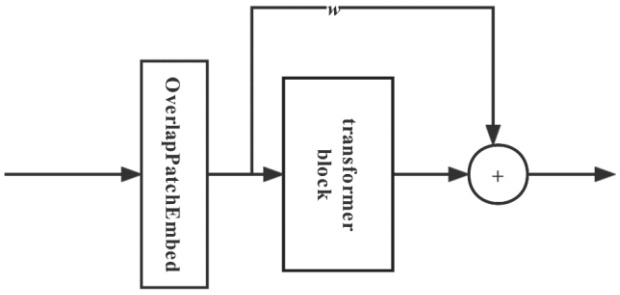


图 3-8 MIT 跳跃结构

MIT 网络遵循 ResNet[67]结构原则：(1)随层网络深度的增加，通道尺寸逐渐增大，空间分辨率逐渐减小。(2)在中间几个阶段会贡献大部分的计算损耗。具体参数细节实现如表 3-2 所示。一共包含有 4 个 Stage，其中每个 Stage 都有一个 OverlapPatchEmbed 层和若干个 Transformer 模块所组成，其中 head 参数代表该 Transformer 模块具有的多头注意力机制的头参数，而其输出分辨率的大小与 OverlapPatchEmbed 的卷积操作相关，当卷积核为 7，步长为 4 时，分辨率变为原输入的 1/4，当卷积核为 3，步长为 2 时，分辨率降为原输入的 1/2。

表 3-2 MIT 网络详细实现细节

	downsp. rate (output size)	MIT-Small	MIT-Base
stage 1	4×	overlap 7,stride 4,64d [dim 64,head 1]×2	overlap 7,stride 4,64d [dim 64,head 1]×3
stage 2	8×	overlap 3,stride 2,128d [dim 128,head 2]×2	overlap 3,stride 2,128d [dim 128,head 2]×4
stage 3	16×	overlap 3,stride 2,320d [dim 320,head 5]×2	overlap 3,stride 2,320d [dim 320,head 5]×6
stage 4	32×	overlap 3,stride 2,512d [dim 512,head 8]×2	overlap 3,stride 2,512d [dim 512,head 8]×3

3.3.2 编解码结构

本文将 MIT 网络与 FairMOT 框架模型中的编解码网络（DLA-Seg 网络）思想相结合，使用 MIT 网络模型强大的特征提取能力，加上多层的 IDA 结构，其中 IDA 模块融合了浅层的底层信息和深层次的语义信息，引入了从浅到深的跳跃连接。形成新的 MIT-Seg 结构，具体网络模型如图 3-9 所示，其中蓝色模块代表 Transformer 模型，作为整个模型的编码器。

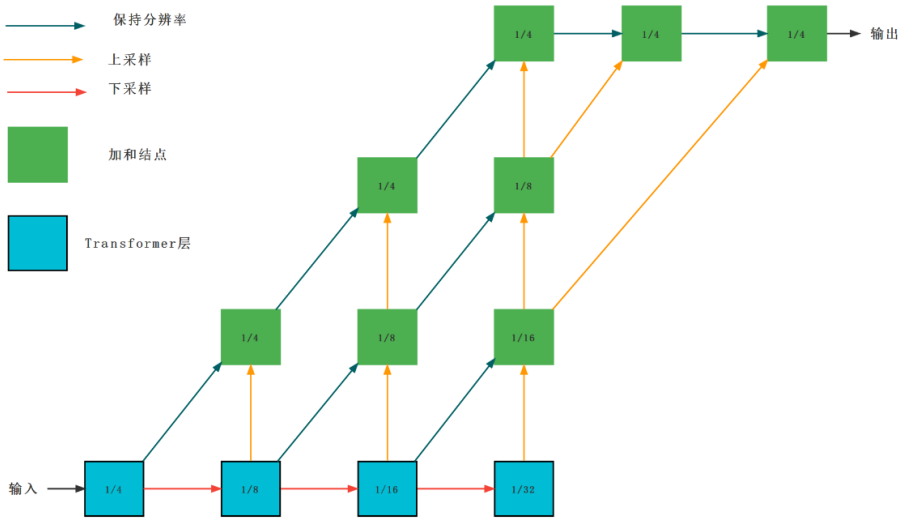


图 3-9 MIT-Seg 编解码网络

通过将 Transformer 编码层提取的特征，使用深层聚合结构来迭代和分层融合特征层次，这种方法使得网络在更少的参数量下获得了更高的准确性。聚合是体系结构的一个决定性方面，随着模块数量的增加，它们的连接性变得更加重要。通过关联聚合通道、规模和分辨率的体系结构，模型确定了对更深层次聚合的需求，并通过迭代的深度聚合和分层的深度聚合来解决这个问题，更有效地利用参数和计算。

3.3.3 基于 Transformer 的解码器优化

解码器网络的设计目标是为了更好地利用骨干网络所提取的特征。对骨干网络所提取的不同阶段的特征图进行再处理和合理使用。通常，一个解码器网络由若干个自底向上的路径和若干个自顶向下的路径所构成。解码器网络是目标跟踪框架中的关键环节，它能大幅度的提高检测性能。

在上一节使用 Transformer 与 DLA-Seg 结构相结合，如表 3-5 所示，取得了略高于基线网络的检测效果，并没有完全发挥出 Transformer 提取特征能力，并且其推理速度和模型复杂度也不适合实时推理，故本文设计一种全新的特征融合编解码网络结构——混合颈部网络（MIX Neck, MN），MN 更适用于 Transformer 骨干网络，减少网络参数的同时使得检测效果进一步提高，由于 FairMOT 很依赖于网络最后一层的特征，所以加上 DLA-UP 模块会涨点特别明显，但是由于网络最后层的输出通道数太多，会增加一些时间损耗，MN 使用 MLP 结构降维，再上采样到融合层中，大大减少了时间损耗。

MN 参考了全卷积神经网络 [68]（Fully Convolutional Networks, FCN）架构和特征金字塔[69]（Feature Pyramid Networks, FPN）自顶向下连接结构，但具体的聚合方法又有所不同，使得模型能够结合不同深度的特征结果实现跳跃链接,这样的操作同时确保模型具有鲁棒性、精确性以及超列特性（对应像素的网络所有节点的激活串联作为特征，进行目标的细粒度定位）。MN 网络结构如图 3-10 所示。网络中的所有卷积操作均使用可形变卷积 DCN 代替，上采样均使用保持分辨率的 3×3 的可形变卷积加上提高分辨率的反卷积操作组成。其中 MLP 结构主要是用于降维操作，使得最终融合层能降低复杂度。最终的 MN 编解码网络结构在 MOT17 验证集上表现高于 FairMOT 基线网络，具体检测评价指标可以查看本章实验验证章节的表 3-6 中实验 4 测试结果。

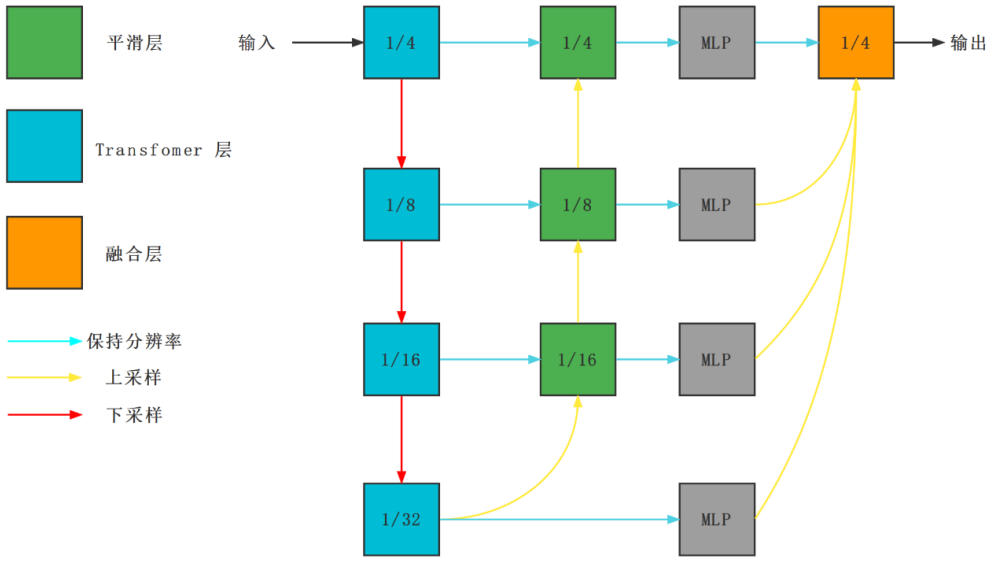


图 3-10 MN 网络

3.4 实验验证

3.4.1 评价指标

（一）基本指标

在目标跟踪任务中很多评价指标和检测和分类相似。GT（Ground Truth）：真实的标签或真实的对象。TP(True Positive)：被标准模型估计为真实的正样本。TN(True Negative)：被标准模型估计为负的负样本。FP(False Positive)：被标准模型估计为正的负样本，也被称为误报。FN(False Negative)：被标准模型估计为负的正样本，也称为漏报。由这几个概念能够推导出如下几个指标。

Accuracy：准确度是指被分类器判定正确的比重，其实就是分类正确的例子占总数的比例,如式（3-13）所示。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3-13}$$

Precision：精确度是指被分类器预测为正的所有样例中真正的正例样本的比重，其计算如式（3-14）所示。

$$Precision = \frac{TP}{TP + FP} \tag{3-14}$$

Recall：召回率是指被分类器预测正确的正例数占预测正例总数的比重，计算如式（3-15）所示。

$$Recall = \frac{TP}{TP + NP} \tag{3-15}$$

AP：Average Precision，AP 就是对 PR 曲线（Recall 值为横轴，Precision 值为纵轴得到的曲线）求积分。实际上，需要对 PR 做平滑处理，对其每个点 Precision 对值取右侧最大的 Precision。

MT：Mostly Tracked trajectories，成功跟踪的帧数占总帧数的 80%以上的 GT 轨迹数量。

ML：Mostly Lost trajectories，成功跟踪的帧数占总帧数的 20%以下的 GT 轨迹数量。

ID switches：因为跟踪的每个对象都是有 ID 的，一个对象在整个跟踪过程中 ID 应该不变，但是由于跟踪算法不强大，总会出现一个对象的 ID 发生切换，这个指标就说明了 ID 切换的次数，指前一帧和后一帧中对于相同 GT 轨迹的预测轨迹 ID 发生切换，跟丢的情况不计算在 ID 切换中。

（二）目标跟踪的综合指标

多目标跟踪精度[70](Multiple Object Tracking Accuracy, MOTA)度量将误报率、误报率和错配率组合成一个单一的数字，为整体跟踪性能提供了一个相当合理的数量。MOTA 是迄今为止最广泛使用的 MOT 评估指标。MOTA 评价指标根据匹配策略，定义了两个非常直观的指标。

多目标跟踪精度(MOTP)[70]，主要反映在确定目标坐标位置上的精度。假设每一帧中有目标$o_1, \dots, o_n$,跟踪器在该帧输出假设为$\{h_1, \dots, h_n\}$，MOPT 可以表示为式（3-16）。

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{3-16}$$

c_t表示第 t 帧目标o_i和假设h_j匹配个数； d_t^i表示第 t 帧目标 o_i 与其匹配的假设坐标位置之间的距离，即匹配误差。MOTP 是在所有帧上对象和与之匹配的假设对象在空间坐标位置上的总误差。

MOTA[70]主要体现在确定目标的数量，以及目标的相关属性方面的准确度，用于统计在跟踪中的误差累积情况，包括 FP、FN、IDs。MOTA 可以表示为式（3-17）。

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{3-17}$$

其中m_t、fp_t和mme_t分别为时间 t 的漏接次数、误报次数和错配次数，计算公式分别为式（3-18），（3-19），（3-20）。

$$\overline{m} = \frac{\sum_t m_t}{\sum_t g_t} \tag{3-18}$$

$$\overline{fp} = \frac{\sum_t fp_t}{\sum_t g_t} \tag{3-19}$$

$$\overline{mme} = \frac{\sum_t mme_t}{\sum_t g_t} \tag{3-20}$$

MOTA 解释了跟踪器在所有帧上造成的对象匹配错误，误报，漏报，不匹配。它提供了一个非常直观的衡量跟踪器在检测对象和保持其轨迹的性能，独立于目标位置估计的精度。

身份识别 F1 分数（Identification F1 Score, IDF1[71]）是针对多目标跟踪模型的跟踪性能评价指标，通过身份随时间的变化计算真实标签和模型输出之间的不匹配的问题。使用假阴 ID 数量（Identification False Negative，IDFN）、假阳 ID 数量（Identification False Positive，IDFP）、真阳 ID（Identification True Positive，IDTP）计数来计算识别精度(IDP)，识别召回(IDR)，以及相应的 F1 评分 IDF1，分别如式（3-21），（3-22），（3-23）。

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{3-21}$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{3-22}$$

$$IDF_1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{3-23}$$

3.4.2 实验环境与训练参数设置

本文实验的硬件环境：Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz，16G 运行内存，2 块 NVIDIA GTX 2080Ti 显卡。软件环境：Ubuntu 18.04.5 LTS，Python 版本为 3.8，Pytorch 版本为 1.7，CUDA 版本为 11.4。

本文使用 AdamW 优化器对模型进行训练，本章实验的所有模型均训练 30 轮，初始学习速率为 10^{-4}。学习率在 20 轮后衰减到 10^{-5}。基线网络以 DLA-34 为骨干网络，使用 DLA-Seg 解码网络作为 Neck 网络，并且加载在 COCO 数据集上预训练的模型权重，批量大小设置为 12，基线网络使用了 FP32 精度训练。由于显存大小的限制在使用 Transformer 为骨干网络时，本文采用了混合精度进行训练即在内存中使用 FP16 精度做存储和乘法计算从而增加训练速度，使用 FP32 做累加避免舍进误差，并且通过放大损失值来防止因为激活梯度值太小而造成的梯度溢出问题。在混合精度训练的情况下，批量大小设置为 8，Transformer 网络使 ImageNet-1K 预训练权重。所有实验均使用标准的数据增强方法，其中包括旋转、缩放和 HSV 颜色通道增强等，并且将输入分辨率固定为 1088×608。

图 3-11 HSV 增强。(a)原图；(b)亮度增强；(c) 色调变化；(d) 饱和度增强

HSV(Hue, Saturation, Value)是基于颜色的直观特性所创建的一种颜色空间, 也被称为六角锥体模型。HSV 模型中颜色参数分别是：色相，饱和度，亮度。HSV

颜色通道增强即通过生成的随机因子来改变这三个数值。HSV 颜色通道增强效果如图 3-11 所示。

全部数据集训练在两台 RTX 2080 GPU 上，由于 CrowdHuman 数据集过于庞大，训练需耗费大量实验资源，本章实验均只用 CrowdHuman 的验证集和 MOT17 一半的数据集进行训练。在 MOT17 的验证集即在一半 MOT17 的数据集上进行验证测试。

因为 CrowdHuman 数据集是为了目标检测任务而提出的数据集，数据集本身并没有没有行人的 ID 信息，并且数据集标签为 COCO 格式，所以本文的实验首先需要使用 ID 生成脚本对 CrowdHuman 数据集进行预处理。具体数据预处理方法：首先将 COCO 格式的数据集转换成 YOLO 数据集格式，在 COCO 格式中，一个边界框由 $[x_{min}, y_{min}, width, height]$四个值所定义，它们是边界框左上角的坐标以及边界框的宽度和高度。而 YOLO 格式中一个边界框由四个值 $[x_{center}, y_{center}, width, height]$表示。其中$x_{center}$和$y_{center}$是包围框中心的归一化坐标。为了使坐标标准化，生成脚本取x_{center}和y_{center}的像素值，它在平面直角坐标系上标记出边界框的中心。然后本文用x_{center}, y_{center}的值除以边界框的宽度$width$的值除以边界框的高度$height$，其中边界框宽度和边界框高度也是经过归一化处理的。最后将图像中的每个对象的 ID 都赋值为-1，这样的操作主要是为了使这部分数据不做身份嵌入特征的损失计算，最终格式如下 $[Frame，\ Identity,\ x_{center},\ y_{center}, width, height]$。

本章具体实验设置为：实验 1，对基线网络使用混合精度训练以及 FP32 全精度训练对比，并且使用新增 GIoU 损失函数与基线网络对比。实验 2，使用基于 Transformer 为骨干网络的两种模型 MIT-Small 和 MIT-Base 与基线网络进行对比。实验 3，在基线网络的基础上不同骨干网络在不同预训练集上纯检测性能对比。实验 4，使用不同的编解码器进行训练，并且冻结身份嵌入特征提取分支参数，从而只进行检测部分训练。

3.4.3 结果对比分析

本章实验的训练结果均在 MOT17 验证集上测试所得。

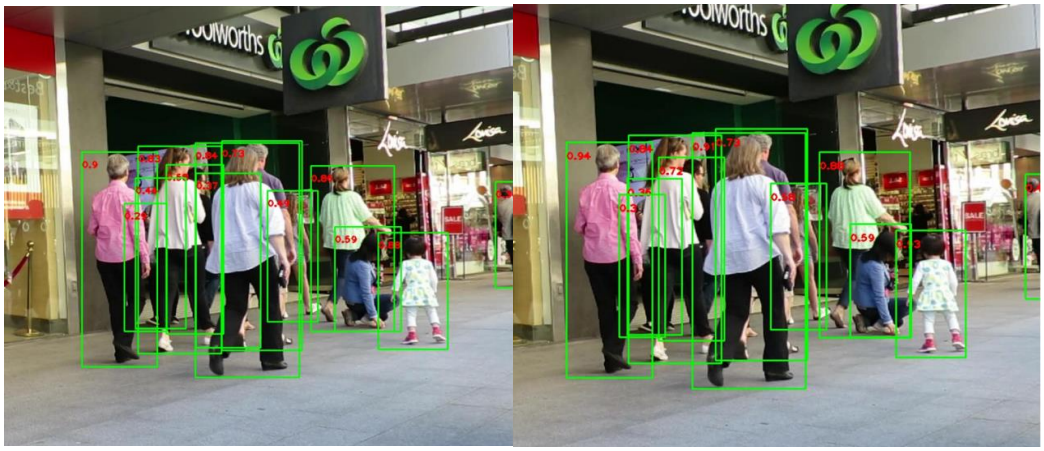
在实验 1 中，在基线网络中分别新增 FP16 混合精度训练和增加 GIoU 损失函数，可以看到最终在 MOT17 验证集进行测试的评价指标对比如表 3-3。使用 FP16 混合精度训练的结果与使用全精度 FP32 训练结果相比，MOTA 指标高了 0.1%，IDF1 指标高了 0.4%，从而可以得出使用混合精度训练的模型并不会比 FP32 全精度训练模型性能差。在使用 FP16 精度下训练，并增加了 GIoU 损失函数后，检测

精度指标 AP 提高 0.06%，并且目标跟踪的评价指标均有提高，其中 MOTA 指标上升了 0.1%，IDF1 上升了 2.5%，IDs 降低到 362。可见新增 GIoU 损失有利于目标跟踪任务，使得目标检测更加精确，从而使得目标跟踪指标上升。

表 3-3 使用混合精度训练和 GIoU 损失函数在 MOT17 验证集上评估指标对比

Backbone	FP16	GIoU	MOTA	IDF1	IDs	AP
DLA34	No	No	69.8	70.2	430	82.32
DLA34	Yes	No	69.9	70.6	430	82.33
DLA34	Yes	Yes	69.9	72.7	362	82.39

新增了 GIoU 损失函数的检测效果与基线网络检测效果对比如图 3-12，本文随机摘取了一帧 MOT17-09 的视频序列帧，很容易的看到增加了 GIoU 损失函数的检测结果，其检测结果的置信度以及边界框的准确度都比基线网络的检测结果更优异。



(a) (b)

图 3-12 GIoU 效果对比。(a)基线损失函数；(b)新增 GIoU 损失函数

将 FairMOT 的骨干网络改成修改后的 MIT 网络，在同样的超参数下，使用混合变压器作为骨干网络在 MOT17 数据集上训练 30 轮，并使用验证集进行验证，可以看到使用 DLA-34 作为骨干网络和 MIT 作为骨干网络的跟踪指标差异如表 3-4。

表 3-4 MOT17 验证集上不同骨干网络的评价指标比较

Backbone	FP16	GIoU	MOTA	IDF1	IDs	Params(M)	FPS(Hz)
DLA34	Yes	Yes	69.9	72.7	362	20.3	22
MIT-Small	Yes	Yes	67.1	73.3	368	20.8	16
MIT-Base	Yes	Yes	68.9	73.6	350	30.8	12

从实验 2 结果表中可以看到使用了 MIT 网络作骨干网络，在目标跟踪的 MOTA 指标上低于基线网络，但在 IDF1 指标上比使用 DLA-34 作为骨干网络的基线网络要高。由此可以得出使用 Transformer 作为骨干网络在检测跟踪任务中依然能取得不错的结果，但使用 DLA-Seg 网络框架加上 Transformer 本身的复杂度使得 FPS 低于原基线网络，在 MIT-Base 作为骨干网络时，参数量远超基线网络，并且 FPS 也远低于基线网络，违背了跟踪实时性的要求，所以在之后的章节中不在使用 MIT-Base 作为骨干网络进行实验。

表 3-5 MOT17 验证集上不同骨干网络及不同预训练集评价指标比较

Backbone	只训练检测网络	预训练数据集	FP16	MOTA	IDF1	IDs	AP
DLA34	Yes	COCO	Yes	70.3	66.8	573	82.60
DLA34	Yes	ImageNet-1K	Yes	67.8	64.6	744	80.85
MIT-Small	Yes	ImageNet-1K	Yes	68.7	65.8	705	81.19

实验 3 对不同骨干网络在相同的环境下冻结其身份嵌入特征提取分支网络仅训练检测网络分支，用来观察其检测能力，实验结果如表 3-5。实验 3 结果表可以看到在通用目标检测的指标 AP 上，基线网络在使用 COCO 预训练权重训练比使用 Transformer 作为骨干网络的追踪器上高 1.41%，但使用 ImageNet-1K 对骨干网络进行预训练的结果比使用 Transformer 作为骨干网络的追踪器低 0.34%，说明 MIT-Small 作为骨干网络在直接使用 DLA-Seg 解码网络的检测能力高于基线网络，但因为基线网络是使用了 COCO 训练集对整个编解码网络进行了预训练，而实验中 MIT-Small 作为骨干网络只使用了 ImageNet-1K 预训练权重并没有让整个编解码网络使用 COCO 训练集预训练，使得 MIT-Small 作为骨干网络的模型的泛化能力更低些，模型很快就过拟合了。

表 3-6 MOT17 验证集上不同解码网络的评价指标比较

Decoder	Backbone	GIoU	冻结 id 头分支	MOTA	IDF1	IDs	AP	FPS(Hz)	Params（M）
DLA-Seg	MIT-Small	Yes	Yes	68.7	65.8	705	81.19	16	20.8M
MN	MIT-Small	Yes	Yes	71.1	70.1	585	82.88	19	19.4M

实验 4 为不同编解码器在 MOT17 验证集上测试结果比较。由表 3-6 中数据可以看到 MN 作为编解码网络比基线网络采用的 DLA-Seg 编解码网络更适合 Transformer 的网络特征融合，并且模型比 DLA-Seg 编解码网络参数量更少，从表

中 FPS（Frames Per Second）可知推理速度上更快，在目标检测的通用指标 AP 上高出基线网络 0.28%，高出同骨干网络为 MIT-Small 的 DLA-Seg 编解码网络 1.69% 。

3.5 本章小结

本章主要提出了一种基于 Transformer 的骨干网络 MIT，并针对骨干网络提出了新的解码器网络 MN，并将其与 DLA-Seg 解码网络进行对比。本章首先阐述了基线网络 FairMOT 算法的网络结构以及跟踪原理，接着分析了该方法中的检测模型的检测能力不足的问题，之后先采用 GIoU 损失函数提高了检测精度，后使用当前特征提取表现优异的 Transformer 网络对该方法进行改进并且重新设计了编解码结构，最后在 CrowdHuman 数据集的验证集和 MOT17 训练集上进行训练，并在 MOT17 验证集上进行测试。测试结果表明本章提出的基于 Transformer 的实时跟踪方法比基线 FairMOT 方法在检测精度上提高 0.28%，并且在只训练检测网络，嵌入特征提取分支使用初始化参数的情况下，跟踪指标 MOTA 提高了 0.8%，IDF1 提高了 3.3%。

第四章 训练策略与匹配策略优化

在第三章本文提出了一种基于 Transformer 的高效实时的行人跟踪算法，验证了 Transformer 用作特征提取的可行性。本章将基于上章所提出方法优化训练策略和匹配策略，从而提高行人跟踪算法的精确度和鲁棒性。

4.1 交替冻结训练策略

单独的行人重识别工作通过学习非常高维度的特征，从而在行人再识别任务领域上得到了很好的结果。然而，本文发现学习低维身份嵌入特征对于一次性多目标跟踪架构实际上是更好的，原因是：多目标跟踪任务中的身份特征匹配与 Re-ID 任务不同。多目标跟踪任务在两个连续视频帧之间只执行少量的一对一匹配。而 Re-ID 任务需要查询大量的候选对象进行匹配，因此需要更多的鉴别性和高维的重识别语义特征，并且高维 Re-ID 特征需要庞大的数据集支持，否者容易过拟合。所以在多目标跟踪任务中并不需要高维的嵌入特征。

多任务学习是多目标跟踪的关键，由于目标跟踪中的身份嵌入特征提取与传统 Re-ID 任务不同，传统 Re-ID 需要与大量的目标进行匹配所以需要更高维和更深层的特征，而目标跟踪只需要与前几帧的目标进行匹配所以不需要那么高维特征，本文最终采用 128 维的特征向量来标识 ID 特征。Transformer 骨干在目标检测任务中表现突出，但是在实际应用到跟踪网络上却并不是很理想，根据第三章中的实验 1 和实验 3 的结果可知，检测精度会受到了目标跟踪中的身份嵌入特征提取任务的影响，由于当前帧的对象只与前 30 帧的对象进行匹配，身份识别所需要的特征比较容易学习到，以至于使用检测特征图就可以获取到身份嵌入所需特征，并且不会因为样本量过少而过拟合。因为目标跟踪任务强依赖检测精度的原因，本文通过训练模型的前 30 轮冻结输出身份嵌入特征分支，只训练检测任务，再冻结检测部分，解冻身份嵌入特征提取分支再训练 10 轮，这样避免了身份嵌入特征提取对目标检测任务的干扰，并且嵌入特征提取直接使用检测任务的特征图进行训练也能得到一个不错的精度，使得检测精度和重识别的精度都能进一步的提高，具体提升可参考本章实验小节的表 4-3 实验 5 的评价指标结果。

4.2 检测框二次匹配策略

检测后跟踪是目前最有效的多目标跟踪方法。由于视频中的复杂场景，检测器容易做出不完美的预测。目前效果优异的多目标跟踪方法[13-17]需要在检测框中

处理真阳性/假阳性权衡，以消除低置信度检测框。然而，消除所有低信度检测框却并不是很好的解决办法，低可信度检测框有时表示物体存在，例如被遮挡的物体。过滤掉这些物体会导致多目标跟踪出现不可逆误差，并带来不可忽略的漏检和碎片轨迹，如图 4-1（b）。

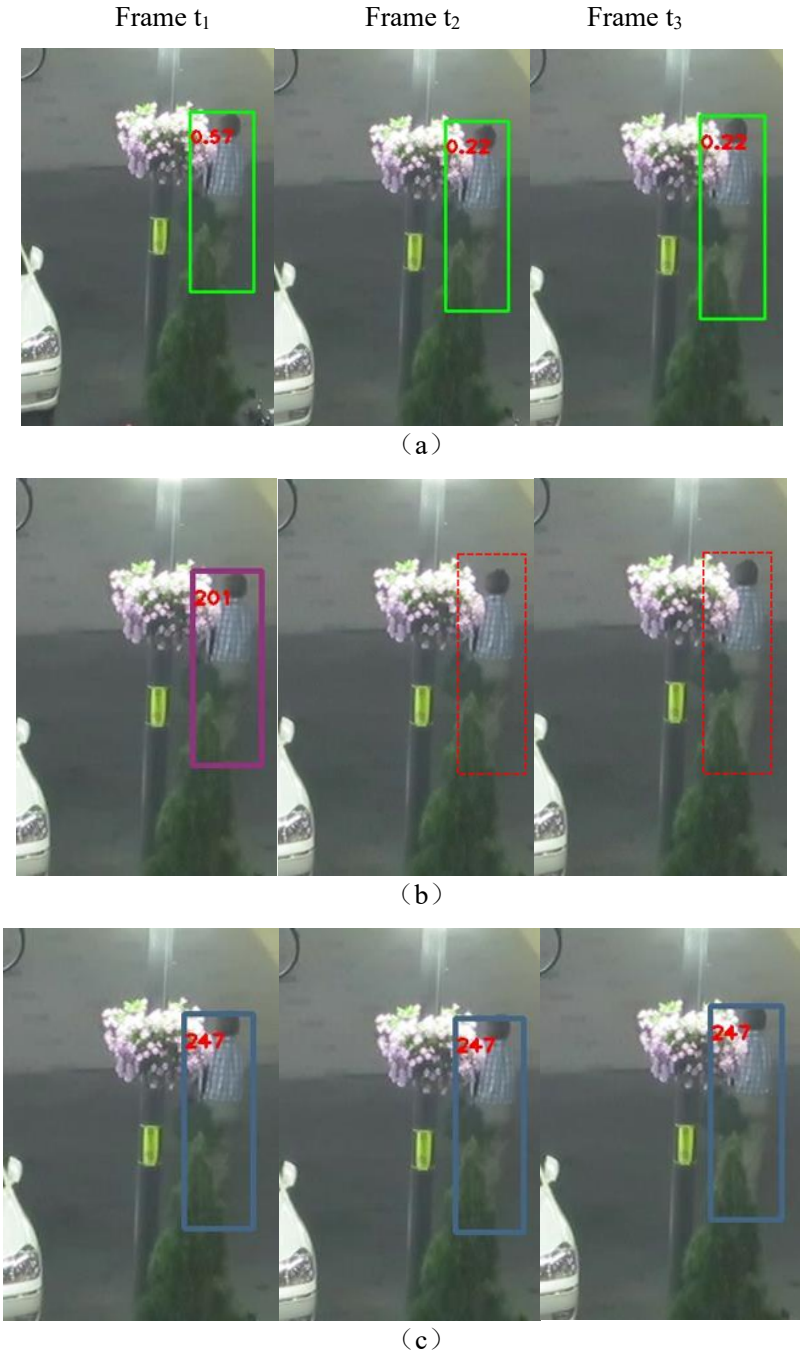


图 4-1 新增低置信度框关联示例。（a）目标检测框结果；（b）使用 FairMOT 单次置信度得到的跟踪框；（c）新增低置信度匹配的跟踪框

本文发现在低分数检测框中，轨迹的相似度提供了一个强有力的线索来区分目标和背景。如图 4-1 (c)所示，低分数检测框通过运动相似度与轨迹匹配，从而正确恢复目标。同时，多余的低置信度检测框作为背景框被移除，因为它没有匹配的轨迹。为了在匹配过程中充分利用高分到低分的检测框，本文采用了一种简单有效的数据关联方法，通过关联每个检测边界框来跟踪。首先使用身份嵌入特征对当前目标进行第一次匹配，再根据运动相似度将高分检测框与轨迹匹配，没有匹配上的高分检测框将会开辟一个新的轨迹。与 SORT 算法类似，使用卡夫曼滤波来预测未来帧中轨迹的位置。运动相似度可以通过预测框和检测框的 IoU 来计算。然后，对不匹配的轨迹进行第二次匹配，即将之前匹配剩下的轨迹与置信度得分较低的检测框进行匹配。这样就可以把由于遮挡而导致的置信度偏低的检测框与之前的轨迹匹配，并去除背景。

与以往的数据关联的方法只保留高分检测框不同，本文将在检测器中检测到的大于一个很低的阈值的检测框都保留下来，并将其分为高分检测框和低分检测框。首先将高分检测框与已经存在的跟踪轨迹关联起来。在这个过程中可能会有一些轨迹与高分检测框无法匹配，因为它没有适当的高分检测框与其匹配。这通常发生在目标被遮挡，运动模糊或目标物体大小变化发生时，因为在这些场景下，检测框的置信度会明显降低，甚至低于设置的高分框的阈值。在高分检测框匹配完成后本文将低分数检测框与这些不匹配的轨迹关联起来，恢复低分数检测框中的对象，同时过滤掉无法与轨迹相匹配的低置信度的检测框，因为这些检测框大多数情况是背景。低置信度匹配的伪代码如表 4-1 所示。

低置信度框二次匹配跟踪匹配算法的输入是一个视频序列 V，以及一个目标检测器 Det 和卡尔曼滤波器 KF。本文还设置了三个阈值t_{high}，t_{low}为检测评分阈值，∈为跟踪评分阈值。新的跟踪算法的输出是视频的轨迹$\mathcal{T}$，每个轨道包含每帧对象的边界框和身份标识。对于视频中的每一帧，算法模型都预测了检测结果边界框和分数。根据检测评分阈值t_{high}和t_{low}，本文将所有检测框分为$\mathcal{D}_{\text{high}}$ 和$\mathcal{D}_{\text{low}}$ 两部分。对于得分高于t_{high}的检测框，匹配算法将其放入高分检测框$\mathcal{D}_{\text{high}}$ 中。对于那些分数范围从t_{low}到t_{high}的检测框，本文将它们放入低分数的检测框$\mathcal{D}_{\text{high}}$（跟踪匹配算法中的第 3 行到第 12 行）。分离低分数检测框和高分数检测框之后,使用卡尔曼滤波器 KF 预测每个跟踪目标的新位置t (跟踪匹配算法中的 13 到 15 行)。第一次关联使用身份嵌入特征，通过嵌入身份特征与所有轨迹中的身份特征通过余弦距离计算相似度，利用匈牙利算法基于相似度完成匹配，匹配的轨迹则会更新其身份嵌入特征和检测框信息，没有匹配的检测框和轨迹将保存在D_{high_1}，$\mathcal{T}_{remain}$之中。

表 4-1 低置信度框二次匹配关联策略伪代码

跟踪匹配算法

输入：视频序列 V；检测器 Det；卡尔曼滤波 KF；检测阈值t_{high}，t_{low}；跟踪阈值∈
输出：视频跟踪结果$\mathcal{T}$

初始化跟踪集：$\mathcal{T} \leftarrow 0$
$for\ \ frame\ f_k\ \ in\ V$ do
  //获取预测检测框和分数
  $D_k, \mathrm{I}_k \leftarrow Det(f_k)$
  $\mathcal{D}_{high} \leftarrow \emptyset$
  $\mathrm{I}_{remain} \leftarrow \emptyset$
  $\mathcal{D}_{low} \leftarrow \emptyset$
  for d, i in $D_k, \mathrm{I}_k\ \ do$
    if $d.score > t_{high}$ then
      $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$
      $\mathrm{I}_{remain} \leftarrow \mathrm{I}_{remain} \cup \{i\}$
    else if $d.score > t_{low}$ then
      $\mathcal{D}_{low} \leftarrow \mathcal{D}_{low} \cup \{d\}$
    end if
  end for
  //预测轨道的新位置
  for t in $\mathcal{T}$ do
    $t \leftarrow KF(t)$
  end for
  //嵌入 ID 匹配
  使用特征余弦距离关联$\mathcal{T}$和I_{remain}
  $\mathcal{D}_{\text{high\_1}}$←来自$\mathcal{D}_{\text{high}}$的剩余对象框
  $\mathcal{T}_{remain}$←$\mathcal{T}$中的剩余跟踪轨迹
  //IOU 首次匹配
  使用 IoU 距离关联$\mathcal{T}$和$\mathcal{D}_{\text{high_1}}$
  D_{remain}←来自$\mathcal{D}_{\text{high_1}}$的剩余对象框
  $\mathcal{T}_{remain\_1}$←$\mathcal{T}_{remain}$中的剩余跟踪轨迹
  //IOU 二次匹配
  使用 IoU 距离关联$\mathcal{T}_{remain_1}$和$\mathcal{D}_{\text{low_1}}$
  $\mathcal{T}_{remain\_2}$←$\mathcal{T}_{remain_1}$中的剩余跟踪轨迹
  //删除不匹配的跟踪轨迹
  $\mathcal{T} \leftarrow \mathcal{T} \backslash \mathcal{T}_{remain\_2}$
  //初始化新轨迹
  for d in D_{remain} do
    if $d.scorce\ >\in$ then
      $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
    end if
  end for
end
返回：$\mathcal{T}$

第二次关联高分检测框$\mathcal{D}_{\text{high_1}}$和跟踪$\mathcal{T}$ (包括失去跟踪$\mathcal{T}_{lost}$)。通过检测框$\mathcal{D}_{\text{high_1}}$与轨迹预测框$\mathcal{T}$之间的IoU计算相似度。然后，利用匈牙利算法完成基于相

似度的匹配操作，匹配的轨迹则更新其嵌入特征和检测框信息。特别地，如果检测框与轨迹框之间的 IoU 小于 0.2，则拒绝匹配。匹配算法保留了D_{high_1}中的不匹配检测框D_{remain}和$\mathcal{T}_{remain}$中的不匹配轨迹$\mathcal{T}_{remain_1}$ (算法 1 中 26 - 27 行)。在低分数检测框$\mathcal{D}_{low}$和第一次关联后的剩余轨迹$\mathcal{T}_{remain_1}$之间进行第二次关联。匹配算法保留$\mathcal{T}_{remain_1}$中未匹配的轨迹，并删除所有未匹配的低分数检测框，因为本文将其视为背景(跟踪匹配算法中的第 22 至 24 行)。

在第二次检测框与轨迹关联中使用 IoU 作为相似度指标是很重要的，因为低分数检测框通常包含严重遮挡或运动模糊，外观特征是不可靠的。关联后，不匹配的轨迹将从轨迹片段中删除。为简便起见，本文在跟踪匹配算法中没有列出轨迹再生的过程。实际上，长期联系是有必要的，以保持轨道的身份嵌入特征，所以本文会对于第二次关联后仍然不匹配的轨迹$\mathcal{T}_{remain_2}$放入$\mathcal{T}_{lost}$中。对于$\mathcal{T}_{lost}$中的每个轨道，只有当它存在的帧数超过一定数量时，即 30 帧，算法才会从轨道$\mathcal{T}$中删除它。否则，保留$\mathcal{T}$中丢失的轨迹$\mathcal{T}_{lost}$ (跟踪匹配算法中的第 32 行)。

最后，在第一次关联后，剩余的不匹配的高分检测框D_{remain}初始化新的轨迹。D_{remain}每个检测框,如果它检测得分高于和存在连续两帧,算法将初始化一个新的跟踪(跟踪匹配算法中的 34 行至 38 行)。每一帧的输出是当前帧中轨迹$\mathcal{T}$的边界框和标识。新增了低置信度二次关联策略后，目标跟踪指标有明显提高，具体见表 4-3 实验 6。

4.3 实验验证

本节主要针对基于 Transformer 的实时跟踪算法的改进算法在 MOT17 验证集进行测试并分析结果。本小节的实验将对所改进的方法进行消融实验，验证其有效性。

4.3.1 训练

在该阶段实验参数仍然沿用第三章的实验设置，使用基于 Transformer 的编解码网络进行提取视频帧的深度特征，采用固定的 1088×608 分辨率的 RGB 彩色图像输入，输出 272×152 的特征图，再通过不同的检测头对特征图进行处理，得到最终的输出。实验 5 使用交替冻结检测分支和嵌入特征提取分支训练策略，经过先冻结嵌入特征提取分支训练 30 轮，再冻结检测分支训练 10 轮。实验 6 使用新的跟踪匹配算法即数据关联部分在 FairMOT 的数据关联算法的基础上新增低置信度检测框的二次匹配策略。实验 7 使用全部数据集（即全部 CrowdHuman 数据集、MOT17 全部数据集、ETH 数据集及 CityPerson 数据集）并在使用之前所有的方法，

在 MOTChallenge 官网上提交结果。由于显存大小限制，基线网络模型训练使用批量大小为12，而使用Transformer作为骨干网络的模型，训练使用的批量大小为8，将初始学习率为$lr = 0.0001$，实验 5 和实验 6 在第 20 个训练周期进行一次 10 倍学习率衰减，实验 7 总共训练 70 个训练周期，并在第 50 个训练周期进行一次 10 倍学习率的衰减，在前 60 个训练周期冻结身份嵌入特征提取分支网络，在第60个训练周期后冻结除身份嵌入特征提取分支以外所有的网络参数。

4.3.2 实验结果

实验 5 使用交替冻结检测分支和嵌入特征提取分支训练策略，经过先冻结嵌入特征提取分支训练 30 轮，再冻结检测分支和编解码网络训练 10 轮，由实验 5 表 4-2 中可以看到，嵌入特征提取任务在对检测任务没有影响下，依然能取得不错的跟踪效果，在使用 MN 解码网络情况下，在无论是跟踪指标还是检测指标均高于基线网络，在跟踪指标 MOTA 和 IDF1 中较实验 4 中结果分别提升了 0.4%，3.4%。可以看到优化后的模型结构的精度无论是 MOTA 指标还是 IDF1 指标都比之前直接使用 DLA-Seg 模型要高。

表 4-2 交替冻结训练策略 MOT17 验证集对比

Decoder	Backbone	MOTA	IDF1	IDs
DLA-Seg	DLA-34	70.1	70.1	515
MN	MIT-Small	71.5	73.5	524

实验 6 在数据关联部分增加了低置信度检测框再匹配策略，从表 4-3 中可以看到目标跟踪评价指标 MOTA，IDF1 均得到了提高了，原 FairMOT 模型在增加了 GIoU 和低置信度检测框再匹配策略后，MOTA 指标提升了 2.3%，IDF1 指标提升了 2.5%，ID 切换次数也降低到 270。而使用 MIX 作为颈部模型，使用 MIT-Small 作为特征提取的骨干网络较基线网络在 MOTA 指标高了 2.6%，在 IDF1 指标上高了 4.0%，ID 切换次数也降低到271。本文方法与基线网络在MOTA验证集中所有序列的 MOTA、IDF1 和 IDs 对比分别如图 4-2，图 4-3，图 4-4。

表 4-3 二次匹配策略评价指标对比

Decoder	Backbone	GIoU	MOTA	IDF1	IDs
DLA-Seg	DLA-34	Yes	72.1	72.5	270
MN	MIT-Small	Yes	72.4	74.2	271

实验7使用了本文的全部的方法并在全数据集上进行训练，并在MOT17测试集上进行测试，然后将测试结果交到 MOTChallenge 官网进行评测，最终得到如表 4-4 的评价结果。

表 4-4 MOT17 测试集评价结果

MOTA ↑	IDF1 ↑	MT ↑	ML ↓	IDs ↓	FPS ↑
74.9	72.5	46.4%	13.6%	3870	18.4

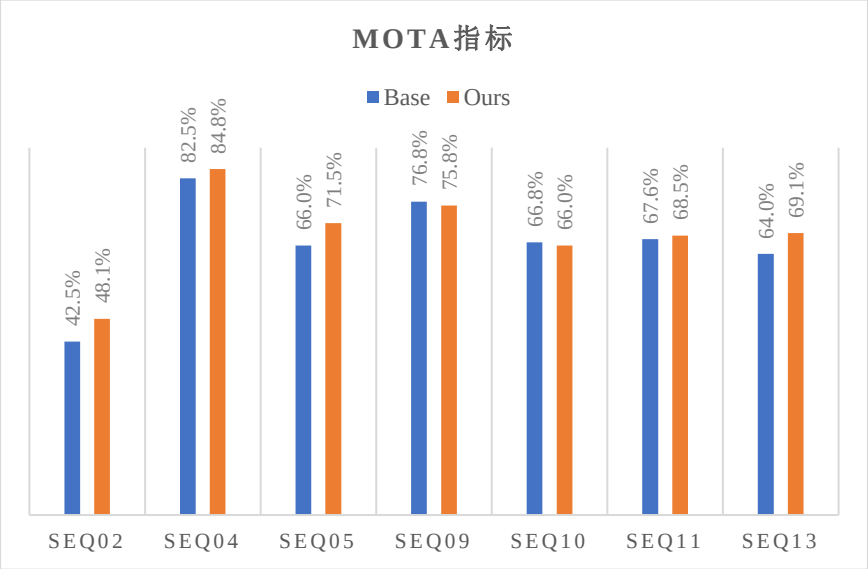


图 4-2 MOT17 视频序列 MOTA 指标对比

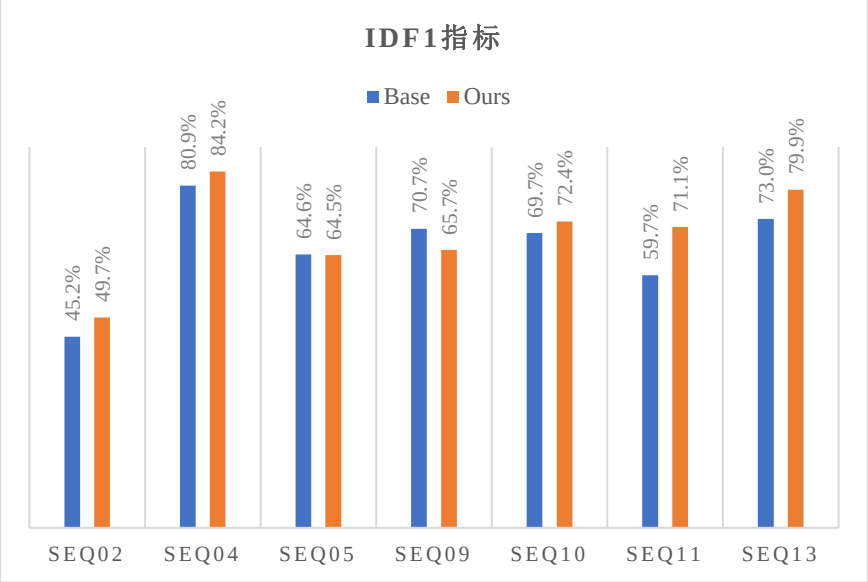


图 4-3 MOT17 视频序列 IDF1 指标对比

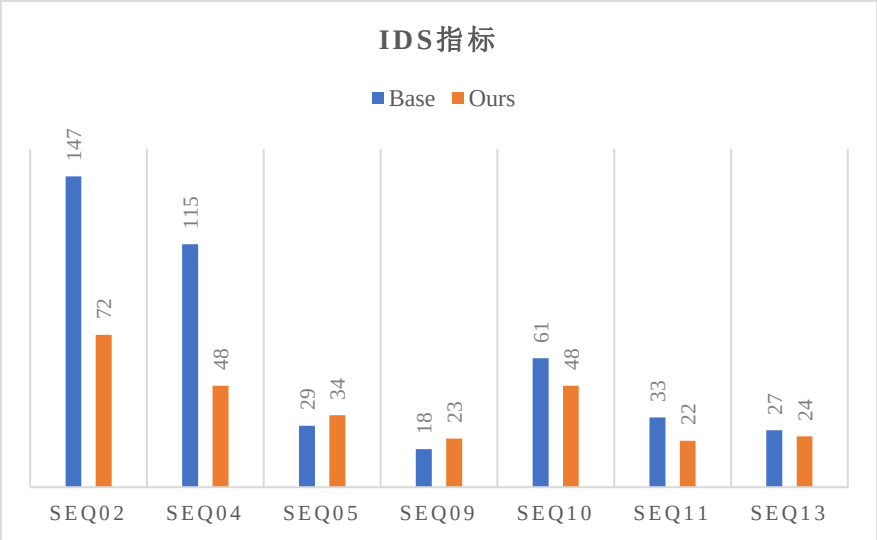


图 4-4 MOT17 视频序列 IDs 指标对比

4.3.3 改进后的效果

表 4-5 比较了使用私人检测器下 MOT17 测试集中近年来表现优秀的方法。FPS 同时考虑了检测时间和关联时间，可以看到本文提出的方法在大部分指标上都优于目前先进的算法。

表 4-5 与目前先进的追踪器对比

Tracker	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	IDs ↓	FPS ↑
DAN[72]	52.4	49.5	21.4%	30.7%	8431	<3.9
TubeTK[73]	63.0	58.6	31.2%	19.9%	4137	3.0
CTrackerV1[74]	66.6	57.4	32.2%	24.2%	5529	6.8
CenterTrack[75]	67.3	59.9	34.9%	24.8%	2898	17.5
QuasiDense[76]	68.7	66.3	40.6%	21.9%	3378	20.3
TransCenter[77]	73.2	62.2	41.8%	18.7%	4614	1.0
FairMOT[25]	73.7	72.3	43.2%	17.3%	3303	**22**
RelationTrack[78]	73.8	**74.7**	41.7%	23.2%	**1374**	8.5
PermaTrackPr[79]	73.8	68.9	43.8%	17.2%	3699	11.9
本文	**74.9**	72.5	**46.4%**	**13.6%**	3870	18.4

图 4-5 显示了本文提出的方法在 MOT17 测试集上的多个跟踪结果。每一行显示了三张按视频序列的时间顺序采样帧的跟踪结果。检测框和 ID 都被标记在图像中，不同颜色的边框代表不同的身份。

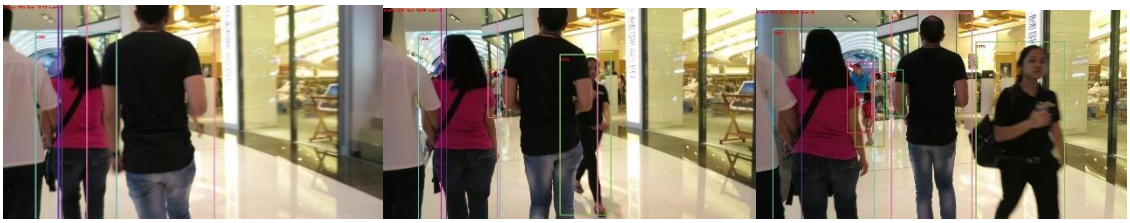
(a)

(b)

(c)

(d)

(e)



(f)

(g)

图 4-5 本文方法在 MOT17 测试集上的实例跟踪结果。(a)MOT17-01；(b)MOT17-03；(c)MOT17-06；(d)MOT17-07；(e)MOT17-08；(f)MOT17-12；(g)MOT17-14

由 MOT17-01 序列帧的结果中可以发现，本文的方法可以在二个行人擦肩而过的时候，通过使用高质量的身份嵌入特征得到正确的身份匹配，而不会导致身份转换，但仅使用 IoU 匹配的跟踪器在这个情形下，通常会出现身份转换。从 MOT17-03 序列帧的结果可以看到，本文的方法即使在高度拥挤的场景下还是可以达到很好的效果。从 MOT17-08 序列帧的结果可以看到，本文的方法在行人被严重遮挡的情况下，既能维持正确的身份标识，又能产出精准的检测框。MOT17-06 序列帧与 MOT17-12 序列帧的试验结果显示，本文的方法有助于解决目标大尺度的变化所带来的不利影响。这主要可以归因于多层特征聚合在网络中的广泛使用。MOT17-07 序列帧和 MOT17 -14 序列帧的检测跟踪结果显示，通过使用本文的方法能够精准地检测出小目标并且维持正确的身份 ID。综上所述，本文所提出的多目标跟踪模型能够在大多数场景下取得非常好的效果。

4.4 本章小结

本章在上章提出的基于 Transformer 的实时跟踪算法的基础之上，提出交替冻结训练策略以及检测框二次匹配策略。首先针对身份嵌入特征提取对检测任务的影响进行了分析，并根据实时跟踪身份嵌入特征所需维度，提出使用检测的特征图作为输入，直接训练身份嵌入特征任务，最终在 MOT17 验证集上目标跟踪指标 MOTA 和 IDF1 较基线网络分别提升了 1.7%，3.3%，证明了身份嵌入提取任务作为次级任务依然能得到很好的跟踪指标。然后在 FairMOT 的跟踪部分新增了检测框的二次匹配策略，将检测框分为高低置信度框进行匹配，最终在 MOT17 验证集上目标跟踪指标 MOTA 和 IDF1 指标较基线网络分别提高了 2.6%，3.5%。最后在本章结尾展示了本文所提出的算法在 MOT17 测试集上所表现出来的效果，比 FairMOT 基线网络在 MOTA 指标上提高了 1.2%，在 IDF1 上提高了 0.2%。

第五章 基于深度学习的行人跟踪系统

本文的第三章和第四章对基于深度学习的行人跟踪方法进行了设计和实验。本章将在前面两章的基础上，基于 Spring、PyTorch 和 Mybatis 开源架构[46]实现了一个轻量级的基于深度学习的行人跟踪视频监控系统，该系统能够利用深度学习模型进行实时视频图像的处理分析，本章将前两章节所训练的行人跟踪算法用于该系统的实际部署。

5.1 实时行人跟踪系统的设计与实现

行人跟踪系统总体运行流程如图 5-1 所示，视频图像采集设备终端实时的采集监控视频数据并进行相应的视频编码之后推流到 Nginx 反向代理服务器上，视频图像处理服务器通过 OpenCV 进行收流，并使用 PyTorch 框架下的算法进行推理，并将推理结果，保存在本地和再次推流到 Nginx 上，后端服务器通过 Flash Player 进行收流，最终显示在网页上。

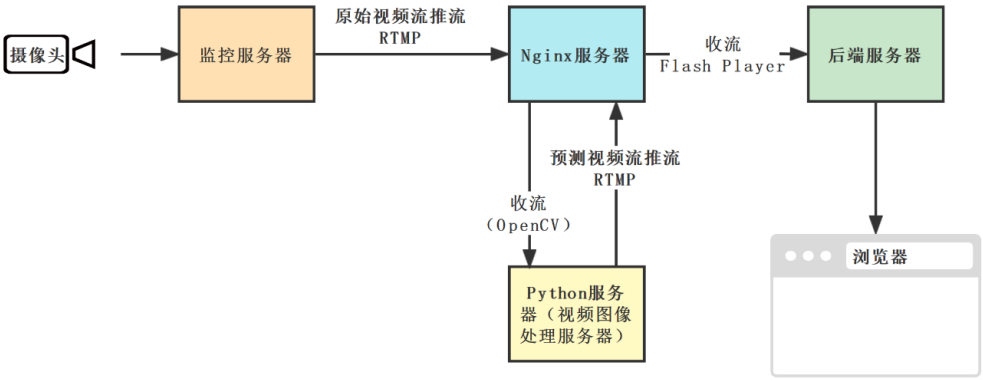


图 5-1 行人系统流程图

本文将行人跟踪视频监控系统分为三个模块，基于 Android 平台的视频图像采集系统，基于 PyTorch 平台的数据处理系统，基于 Web 的展示系统，如图 5-2 所示。基于 Android 平台的视频图像采集系统将安卓的摄像设备所产生的视频流转成实时消息传输协议（Real Time Message Protocol，RTMP）推流到 Nginx 服务器。基于 PyTorch 平台的数据处理系统主要任务是将从 Nginx 反向代理服务器拉流并将其转换成视频帧，通过调用算法模型并使用 GPU 加速设备运算，最终得到推理结果，并且将结果绘制到原视频帧图像上，最后将视频帧转成预测比特流回

推给 Nginx 服务器。基于 Web 的展示系统的主要任务是通过 Flash 播放器对预测流进行收流，在前端页面上进行展示，除此之外，还增加了实时人数统计，时段人流统计等功能实现。

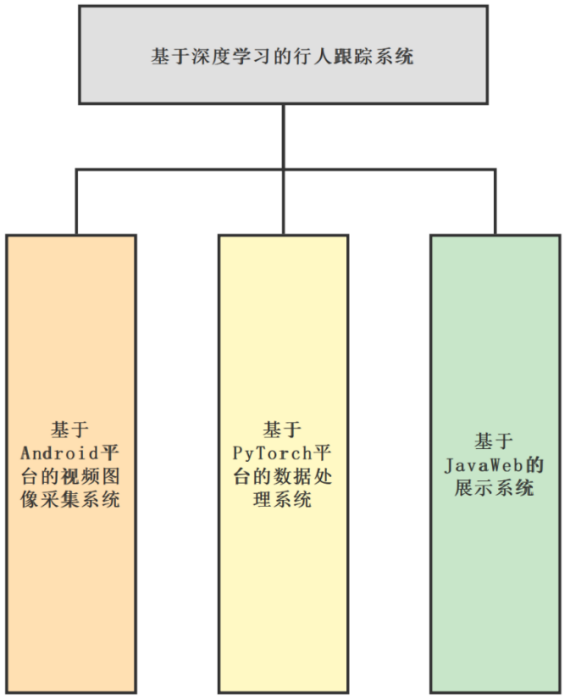


图 5-2 系统模块示意图

5.2 基于 Android 平台的视频图像采集系统

基于 Android 平台的视频图像采集系统主要功能是采集视频并将视频推流给服务器。主要由三个部分组成，分别是采集、编码封装和推流。

采集是图像采集系统的第一个环节，从 Android 系统摄像设备中获取原始视频数据，而视频采集设计两方面数据的采集：音频数据采集和图像采集。音频的采集过程主要通过设备将环境中的模拟信号采集成 PCM 编码(Pulse Code Modulation, 脉冲编码调制)的原始数据，图像的采集过程主要由摄像头等设备拍摄成 YUV 编码（Y 代表的是亮度，UV 代表的是彩度）的原始数据。

编码和封装是图像采集系统的第二个环节，主要是对采集到的原始数据进行处理，使其能够通过压缩，提高传输效率。本文使用 JavaCV 中的 FFmpeg 编码器进行编码。JavaCV 是由多种开源的计算机视觉库所组成的包装库，提供了许多实用程序类，使其功能更容易在 Java 平台上使用，包括 Android。JavaCV 开源库中的 FFMpegFrameRecoder 类将原始视频数据压缩成 H.264 编码再封装成 FLV（Flash Video）格式。

推流是图像采集系统的最后环节，本文采用 H.264 编码对视频流进行编码，主要是因为基于超文本传输协议（Hyper Text Transfer Protocol，HTTP）的自适应码率流媒体传输协议（HTTP Live Streaming，HLS）要求使用这种编码格式。主要的推流协议有 RTMP 和 HLS 两种，具体区别见表 5-1。

表 5-1 RTMP 与 HLS 区别

	RTMP	HLS
公司	Adobe	Apple
平台支持	支持 Flash Player 的网页端，Vitamio，移动端。	安卓（版本 3.0 以上），苹果产品，桌面浏览器需要三方库。
延迟	3s 左右延迟，实时性较高。	根据传输流长度而定，平均延迟 10s。

RTMP 协议基于 TCP（Transmission Control Protocol，传输控制协议），是一种设计用来进行实时数据通信的网络协议，主要用来在 Flash/AIR 平台和支持 RTMP 协议的流媒体/交互服务器之间进行音视频和数据通信。根据系统目标跟踪的实时性要求，本文选择 RTMP 作为推流协议。

数据采集系统的具体流程如图 5-3，首先使用安卓应用程序接口（Application Programming Interface，API）自带的相机接口，实现从摄像头采集图像。然后是 JavaCV 开源库中的 FFMpegFrameRecorder 类实现对相机采集到的帧编码并进行推流。

图 5-3 安卓端收推流流程图

5.3 基于 PyTorch 平台的数据处理系统

基于 PyTorch 平台的数据处理系统主要功能有拉取视频流并将其转为视频帧，使用深度学习模型进行推理计算。其执行流程如图 5-4 所示。

整个数据处理系统主要分为 4 个线程执行，包括主线程，数据采集线程，模型推理线程，数据分发线程。其中主线程主要用作参数配置、共享变量设置以及推流管道设置。

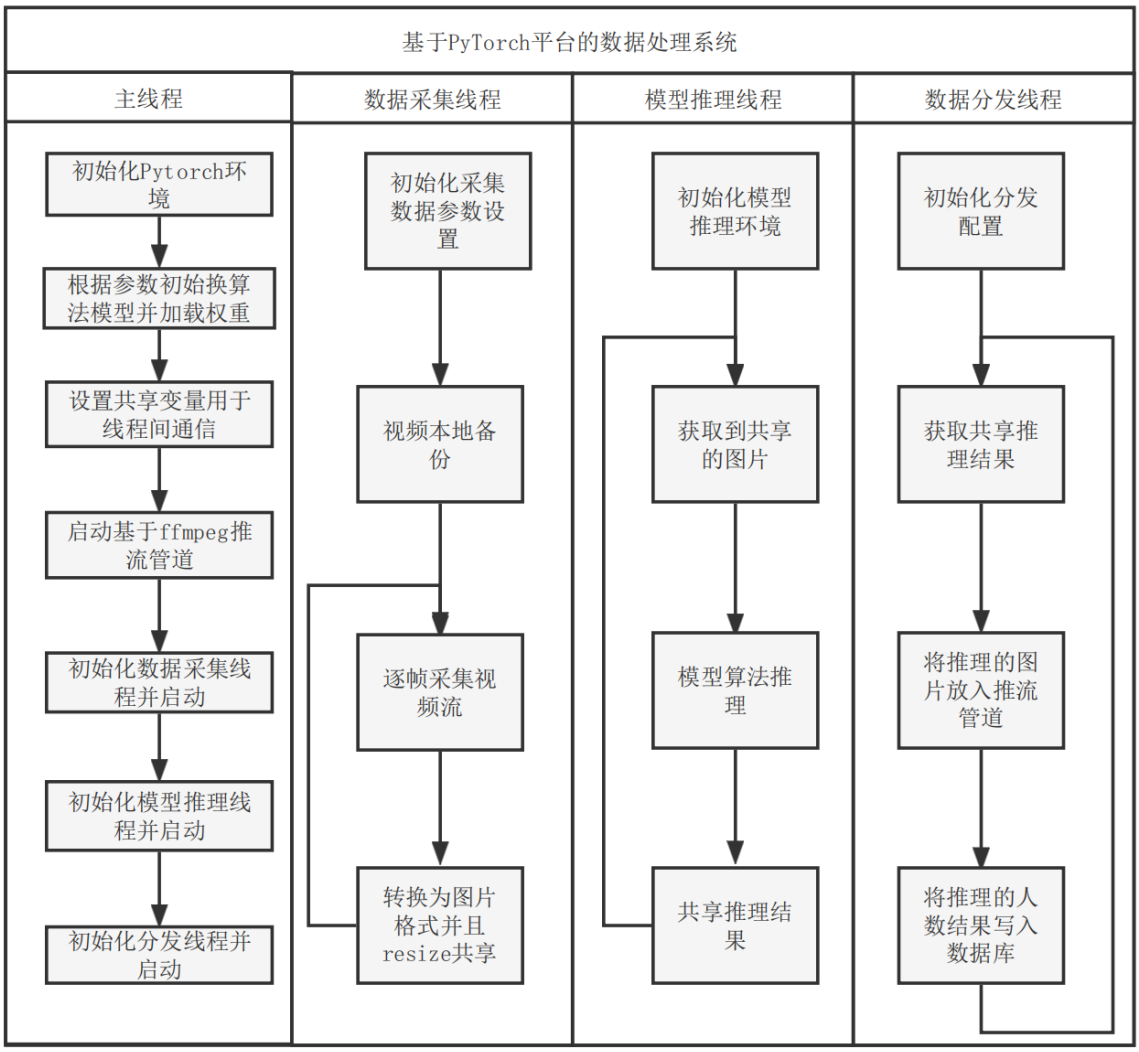


图 5-4 基于 Pytorch 平台的数据处理系统

OpenCV-Python 是原始 OpenCV 实现的 Python 包装器，原始 OpenCV 是由 C++实现的。OpenCV-Python 使用 NumPy（Numerical Python），这是一个高度优化的数据操作库。所有 OpenCV 数组结构都转换为 NumPy 数组。这也使得与使用 NumPy 的其他库集成更容易。

在数据采集线程中通过 OpenCV-Python 的 VideoCapture 类将远程的视频流与对象进行绑定，使用 read()成员函数返回远程视频流的每一帧的元组对象，使用 PIL（Python Image Library）对每一帧的元组对象进行处理，使其还原为 RGB 颜色系统的图像格式。

模型推理线程则通过共享变量获取到 RGB 图像数据，而 RGB 颜色模型是数字图像中比较常见的一种颜色模型，能够反映此时此刻图像中 RGB 的分布。但是

在实际图像处理中，往往会碰到一个比较头疼的问题，就是图像的 RGB 容易受光照的影响，也就是说，同一位置，不同光照强度会导致图像的 RGB 值发生很大变化，并且模型推理精度和速度会直接受到图片尺寸大小影响。所以本文模型推理线程首先将输入图像进行调整大小，再对图像进行归一化操作，RGB 归一化公式如式（5-1），（5-2）和（5-3）。

$$r = \frac{R}{R+G+B} \tag{5-1}$$

$$g = \frac{G}{R+G+B} \tag{5-2}$$

$$b = 1 - \mathrm{r} - \mathrm{g} \tag{5-3}$$

随后将图片输入到已经训练好的模型当中，模型的推理结果数据将会返回到主线程的共享变量里，其中包括推理图像以及当前帧人数统计。

数据分发线程，会使用 FFmpeg 子进程将转化为 byte[]编码格式的数据封装成 FLV 格式。并且会定时调用基于 JavaWeb 的展示系统的后端服务器，将人流量通过后端接口存在 MySQL 里行人人数统计表中。

5.4 基于 Web 页面的展示系统

基于 Web 页面的展示系统主要工作是对上节数据处理系统所传输的数据进行展示。本节使用了 SSM（Spring, SpringMVC, MyBatis）框架对网页的前后端进行设计与实施。展示系统主要分为三个页面，登录与注册页面，实时视频流展示页面，数据分析页面。

图 5-5 登录与注册页面

登录与注册模块，页面如图 5-5 所示，所有用户的账号信息都将会被存在 MySQL 数据库中。数据库中的登录与注册信息表主要包含账户 ID、用户名、注册密码、密码盐值、注册邮箱、账号激活状态、用户注册时间和用户访问权限。在 Web 应用中必须考虑用户信息安全性的问题，为了避免用户的账号密码被窃取，所以用户的密码不能以明文的方式存在数据表中，必须对密码进行加密操作。本文在用户信息表中增加一列存储盐值，使得本文存在 SQL 表中的密码为用户的输入密码与盐值通过 MD5 信息摘要算法（MD5 Message-Digest Algorithm，MD5）加密之后的字符串。由于 HTTP 协议具有无状态性，为了升级用户体验，确保用户能够正常使用网站的功能而非每次访问都需要重新输入用户账号和密码，本文单独建立了一张表用以存储用户的登录信息。在验证用户名和密码时，验证成功后服务端会给客户端下发一个认证票据（Ticket），通过 Cookie 发送给客户端，最终储存在用户本地终端上，在之后的客户端的每次请求都将带上票据与数据库比对，并且使用定期过期的策略，做到每隔一段时间需要重新验证用户信息。

实时视频流展示模块，主要用于展示实时行人跟踪监控视频，并显示行人数量，页面如图 5-6 所示。通过 Flash 播放器可以轻松做到实时收流并播放，使用定时轮询行人表，从而做到实时行人数量更新。在视频展示页面，用户可以选择视频源，本系统支持多源，在用户选择了视频源后，实时监控视频模块会自动播放推理的监控源。在视频中行人会被不同颜色的边界框标记，并为每个行人分配专属的 ID，并在右方控制台可以看到进站观看视频的访客人次和视频中目标人数。

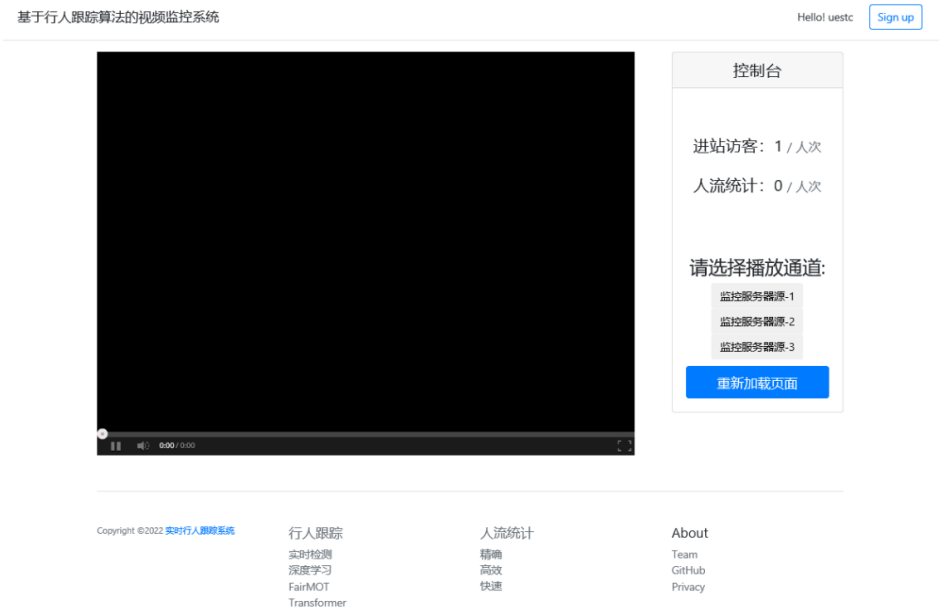


图 5-6 视频展示页面

图 5-7 数据可视化页面

数据分析模块，通过每日的人流统计，可以轻松统计出每日的人流量，通过柱状图展示到数据分析页面，在每日统计的基础上，增加了每月统计，将每月人流量占比可视化。数据可视化如图 5-7 所示。用户可以通过数据可视化页面的查看人流日维度统计的柱状图和月维度的饼状图。这样用户可以清晰的看到一个月的人流分布情况，整年的人流占比情况。

5.5 系统部署及测试

在开发中，由于行人跟踪系统的三个子系统都是独立运行的，在联调时往往会出现一些系统漏洞，为了确保系统的正常运行，需要对系统中的功能进行相关测试，以及对系统软硬件要求进行测试。为了使深度学习模型能够实时推理并且后端服务器能够正常运行，系统的实际线下部署需要满足如表 5-2 中所列出的软硬件要求。

表 5-2 行人跟踪系统的硬软件要求

项目	要求
操作系统	Ubuntu 18.4 x64
CPU	主频 2.0GHz 以上
内存	8G 以上
GPU	NVIDIA GeForce GTX1060（6G）以上
CUDA 版本	10.2
算法框架	PyTorch 1.7
开发语言	Python3.8 、JDK1.8
反向代理 web 服务器	Nginx with RTMP
数据库	MySQL8.0
视频处理	FFMPEG

本文将 PyTorch 平台的数据处理系统与基于 Web 页面的展示系统都部署到同一个服务器上。之后进行了线上联调并且验证了系统能否正常工作，结果显示是否符合预期。具体验证如下内容：基于 Android 平台的视频图像采集系统推流是否正常、基于 PyTorch 平台的数据处理系统是否能够拉到视频流、基于 PyTorch 平台的数据处理系统是否能够正常的处理视频流并推回给 Nginx 服务器、Java 后端能否取到和修改数据库中的数据、前端收流是否正常、网页用户注册和登录是否正常、页面是否正常显示等。

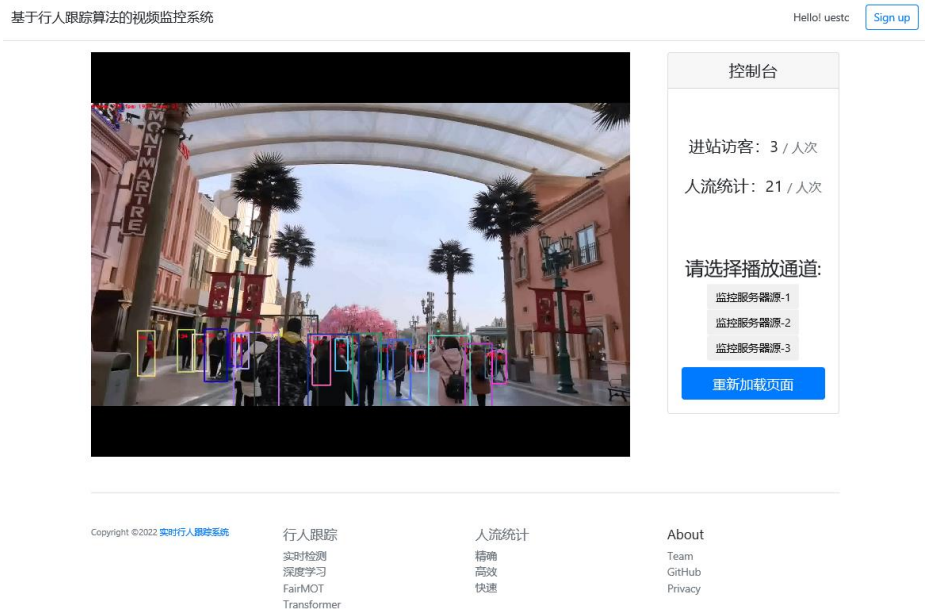


图 5-8 行人跟踪系统运行图

经过验证，本文提出的基于深度学习的行人跟踪系统可以成功完成视频采集，视频推理，数据分析三个功能。行人跟踪系统运行时主页面如图 5-8 所示，可以在监控视频中看到检测框和行人身份标识都被标记在图像中，不同颜色的边框代表不同的身份，并且在左上角显示了监控视频的实时帧数。本系统只是简易的行人跟踪系统，后续可以根据具体的业务需求而对系统进行扩展，比如异常行为检测，安全帽检测，口罩检测等等。

5.6 本章小结

本章将基于深度学习的行人跟踪模型应用于实际部署，设计了基于实时监控的行人跟踪系统。本章分别对基于 Android 平台的视频图像采集系统，基于 PyTorch 平台的数据处理系统，基于 Web 页面的展示系统三个子系统的设计与实现进行介绍。系统主要实现了一个基于深度学习的行人跟踪系统，通过使用预先训练好的行人跟踪算法权重，对实时采集的视频数据进行推理，最终展示给用户。

第六章 总结与展望

6.1 全文总结

行人跟踪一直是计算机视觉领域的一个需要长期研究任务，本文以统一检测网络和身份嵌入特征提取网络的多目标跟踪框架为研究背景，主要对以下四个方向进行研究：

（1） 使用 Transformer 模型作为行人跟踪的骨干网的可行性进行探索，进而修改编解码网络结构适应 Transformer 的特征提取，提出一种更轻量的编解码结构 MN，实验表明，本文提出的 MN 网络结构比基线网络所使用的 DLA-Seg 网络在检测精度上有了明显的提高，在通用目标检测评价指标 AP 上提高了 0.2%。

（2） 对目标跟踪所需的身份嵌入特征提取进行研究，提出了交替冻结训练策略。使用纯检测特征作为身份嵌入特征提取分支的输入能够避免身份嵌入特征提取任务对检测任务的影响，并且由于行人跟踪任务中对重识别范围只针对前几帧的特征进行匹配判定，所以他们并不需要太多的维数就能达到饱和的精度，在经过交替冻结检测分支和身份嵌入特征提取分支训练后，本文的网络获得更高的检测精度和更好的嵌入特征，比基线网络的跟踪指标 MOTA 指标增加了 1.7%，IDF1 提高了 3.3%。

（3） 本文在数据关联部分新增了二次匹配策略，即低置信度检测框的召回方法，通过将低置信度的推理框与未匹配上的轨迹进行匹配，提高跟踪精度，减少身份切换，在这个方法下，IDF1 在原有改进的基础上指标再次提高了 0.7%，MOTA 指标提高了 0.9%。

（4） 在整个网络优化完成后，本文还对基于深度学习的行人跟踪系统进行了设计和实现，最终通过 SSM 框架实现了一个简易的行人跟踪系统，使用安卓设备进行实时拍摄推流，通过 Nginx 服务器对直播流进行转发，使用 Python 服务器进行推理，最后在 Web 页面进行展示。本文提出的行人实时跟踪检测系统完成实时行人跟踪，实时人数统计，并增加人流量数据分析功能。

6.2 后续的工作展望

Transformer 方法在行人跟踪领域的研究近几年发展迅速，但对于任务实时性还没有进行深入得研究。在本文研究工作的基础上，仍有以下几个方向值得进一步研究：

（1） 本文还是基于当前帧进行目标的检测，缺少时序上关联，可以给模型增加时序模块，使整个模块能够增加上下帧关联增加检测的精度，从而增加跟踪精度，减少身份标识的切换。

（2） 本文跟踪模型的骨干网络是在 ImageNet-1K 数据集上进行预先训练，如果能将检测模型的编解码网络在 COCO 训练集上进行预训练，则模型的泛化能力能进一步的提高。

（3） 本文模型使用了 Transformer 网络作为特征提取器，但整体框架还是使用了卷积网络进行解码，可以尝试 Transformer 直接获取到 ID 和位置信息，减少中间计算层。

（4） 本文部署还是使用的原模型权重进行推理，后续可以考虑对模型进行轻量化操作，如量化和蒸馏。蒸馏即训练比较大型的网络模型用以蒸馏训练较为轻量的模型使轻量化的模型能够达到大模型的推理精度。量化即将原来的模型的数据字符大小转换为更低空间大小数据格式，例如对模型进行 INT8 量化，在量化为 INT8 数据大小后，整体模型会减少约 4 倍的内存大小，推理速度会提高 2-3 倍，更适合部署。

（5） 本文提出的行人跟踪系统只有简易的行人跟踪和行人数据统计功能，后续可以对系统进行扩展，比如增加异常行为识别功能，异常行为报警，截图本地保存功能等。

致 谢

在电子科技大学的校园生活转眼间就要过去，回想起在攻读硕士学位的这三年里，我经历了许多有趣的故事，认识了许多志同道合的朋友，也掌握了不少实用的技能。我很庆幸能够在电子科技大学这所巍巍学府中遇见了这么好的导师和同学，和导师和同学相处中让我学习到了为人处世之道和拼搏向上的精神。在这里我的眼界变得开阔，思想变得深邃。

时光荏苒,岁月如梭，在这临近离别之际，首先我要衷心感谢我的导师漆进。对于离开校园已经工作并且不是本专业的学生，是漆进老师给了我重新返回学校跨专业学习的机会。刚开始我还是一个跨专业，零基础的小白，在漆老师的悉心教导之下，逐渐成长，在图像和视频处理，深度学习领域也有一些自己的理解。感谢漆老师的栽培，在这里祝漆老师及其家人，在往后的日子中能够工作顺利，平安健康，幸福美满。

接着我要感谢我的家人，特别是我的妻子，是他们一直支持我的学习和生活，他们尊重我的每一个选择，让我能够放下负担去做我自己想做的事情。没有他们的支持，或许我还停留在狭小的圈子之中，更不会想到有一天自己也能成为别人家的孩子，谢谢他们，希望在以后的日子家人们都要平安幸福。

之后要感谢我的朋友，特别是我的室友们，他们像是路灯一样指引着我前进，在这三年里，无论是学习还是生活都给了不少建议，让我受益匪浅。正是他们的陪伴之下，使得我这三年才能过得这么绚丽多彩。尽管之后我们可能不在一个地区，将散落到天南海北，希望我们友谊长存，并且前程似锦。

最后感谢细心负责的辅导员，为了我们的学习和生活都操碎了心，时时提醒我们每个阶段需要做的事情，并在许多事情上为我们排忧解难。感谢所有教授我专业知识的老师和评阅论文并给论文提出宝贵的指导意见的专家。

参考文献

[1] Luo W, Xing J, Milan A, et al. Multiple object tracking: a literature review[J]. Artificial Intelligence, 2021, 293: 103448.

[2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems (NIPS), 2017, 30.

[4] Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking[C]. IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016: 3464-3468.

[5] 彭丁聪.卡尔曼滤波的基本原理及应用[J].软件导刊,2009,8(11):32-34.

[6] 柳毅,佟明安.匈牙利算法在多目标分配中的应用[J].火力与指挥控制,2002(04):34-37.

[7] Bochinski E, Eiselein V, Sikora T. High-speed tracking-by-detection without using image information[C]. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 2017: 1-6.

[8] Bae S H, Yoon K J. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning[C]. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014: 1218-1225.

[9] Xiang Y, Alahi A, Savarese S. Learning to track: online multi-object tracking by decision making[C]. IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015: 4705-4713.

[10] Zhang L, Li Y, Nevatia R. Global data association for multi-object tracking using network flows[C]. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008: 1-8.

[11] Berclaz J, Fleuret F, Turetken E, et al. Multiple object tracking using k-shortest paths optimization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(9): 1806-1819.

[12] Milan A, Roth S, Schindler K. Continuous energy minimization for multitarget tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 36(1): 58-72.

[13] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]. IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017: 3645-3649.

[14] Yu F, Li W, Li Q, et al. Poi: Multiple object tracking with high performance detection and appearance feature[C]. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016: 36-42.

[15] Mahmoudi N, Ahadi S M, Rahmati M. Multi-target tracking using cnn-based features: cnnmtt[J]. Multimedia Tools and Applications, 2019, 78(6): 7077-7096.

[16] Zhou Z, Xing J, Zhang M, et al. Online multi-target tracking with tensor-based high-order graph matching[C]. International Conference on Pattern Recognition (ICPR), Beijing, China, 2018: 1809-1814.

[17] Fang K, Xiang Y, Li X, et al. Recurrent autoregressive networks for online multi-object tracking[C]. IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 2018: 466-475.

[18] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[19] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[20] Kokkinos I. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017: 6129-6138.

[21] Ranjan R, Patel V M, Chellappa R. Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 41(1): 121-135.

[22] Voigtlaender P, Krause M, Osep A, et al. Mots: multi-object tracking and segmentation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019: 7942-7951.

[23] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017: 2961-2969.

[24] Wang Z, Zheng L, Liu Y, et al. Towards real-time multi-object tracking[C]. European Conference on Computer Vision (ECCV), Glasgow, US, 2020: 107-122.

[25] Zhang Y, Wang C, Wang X, et al. Fairmot: On the fairness of detection and re-identification in multiple object tracking[J]. International Journal of Computer Vision, 2021, 129(11): 3069-3087.

[26] Zhou X, Wang D, Krähenbühl P. Objects as points[J]. arXiv preprint arXiv:1904.07850, 2019.

[27] Feichtenhofer C, Pinz A, Zisserman A. Detect to track and track to detect[C]. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017: 3038-3046.

[28] Luo H, Xie W, Wang X, et al. Detect or track: towards cost-effective video object detection/tracking[C]. AAAI Conference on Artificial Intelligence, Hawaii, USA, 2019, 33(01): 8803-8810.

[29] Tang P, Wang C, Wang X, et al. Object detection in videos by high quality object linking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 42(5): 1272-1278.

[30] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[31] Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention[C]. International Conference on Machine Learning. PMLR, 2021: 10347-10357.

[32] Yuan L, Chen Y, Wang T, et al. Tokens-to-token vit: training vision transformers from scratch on imagenet[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, Canada, 2021: 558-567.

[33] Chu X, Tian Z, Zhang B, et al. Conditional positional encodings for vision transformers[J]. arXiv preprint arXiv:2102.10882, 2021.

[34] Han K, Xiao A, Wu E, et al. Transformer in transformer[J]. Advances in Neural Information Processing Systems, 2021, 34.

[35] Chen C F R, Fan Q, Panda R. Crossvit: cross-attention multi-scale vision transformer for image classification[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 357-366.

[36] Li Y, Zhang K, Cao J, et al. Localvit: bringing locality to vision transformers[J]. arXiv preprint arXiv:2104.05707, 2021.

[37] Wang W, Xie E, Li X, et al. Pyramid vision transformer: a versatile backbone for dense prediction without convolutions[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 568-578.

[38] Liu Z, Lin Y, Cao Y, et al. Swin transformer: hierarchical vision transformer using shifted windows[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 10012-10022.

[39] Wu H, Xiao B, Codella N, et al. Cvt: introducing convolutions to vision transformers[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 22-31.

[40] Xu W, Xu Y, Chang T, et al. Co-scale conv-attentional image transformers[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 9981-9990.

[41] Graham B, El-Nouby A, Touvron H, et al. LeViT: a vision transformer in convnet's clothing for faster inference[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 12259-12269.

[42] Chu X, Tian Z, Wang Y, et al. Twins: revisiting the design of spatial attention in vision transformers[J]. Advances in Neural Information Processing Systems, 2021, 34.

[43] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]. European Conference on Computer Vision (ECCV), Glasgow, US, 2020: 213-229.

[44] Kalyanaraman A, Griffiths E, Whitehouse K. Transtrack: tracking multiple targets by sensing their zone transitions[C]. International Conference on Distributed Computing in Sensor Systems (DCOSS), Washington, DC, USA, 2016: 59-66.

[45] Meinhardt T, Kirillov A, Leal-Taixe L, et al. Trackformer: multi-object tracking with transformers[J]. arXiv preprint arXiv:2101.02702, 2021.

[46] 徐雯, 高建华. 基于 Spring MVC 及 MyBatis 的 Web 应用框架研究[J]. 微型电脑应用, 2012, 28(7):5.

[47] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: a metric and a loss for bounding box regression[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019: 658-666.

[48] Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning[J]. stat, 2016, 1050: 23.

[49] Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks[C]. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017: 764-773.

[50] Zhu X, Hu H, Lin S, et al. Deformable convnets v2: more deformable, better results[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019: 9308-9316.

[51] Zoph B, Le Q V. Searching for activation functions[C]. International Conference on Learning Representations, Vancouver(ICLR), Canada, 2018: 1-13.

[52] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017: 2980-2988.

[53] Robbins H, Monro S. A stochastic approximation method [J]. Annals of Mathematical Statistics, 1951, 22(3):400-407.

[54] Kingma D P, Ba J. Adam: a method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[55] Loshchilov I, Hutter F. Fixing weight decay regularization in adam[J]. arXiv preprint arXiv:1711.05101,2017.

[56] Yu F, Wang D, Shelhamer E, et al. Deep layer aggregation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018: 2403-2412.

[57] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014: 580-587.

[58] Girshick R. Fast r-cnn[C]. IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015: 1440-1448.

[59] Cai Z, Vasconcelos N. Cascade r-cnn: delving into high quality object detection[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018: 6154-6162.

[60] Neubeck A, Van Gool L. Efficient non-maximum suppression[C]. International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 2006, 3: 850-855.

[61] Liu W, Anguelov D, Erhan D, et al. Ssd: single shot multibox detector[C]. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016: 21-37.

[62] Milan A, Leal-Taixé L, Reid I, et al. MOT16: a benchmark for multi-object tracking[J]. arXiv preprint arXiv:1603.00831, 2016.

[63] Shao S, Zhao Z, Li B, et al. Crowdhuman: a benchmark for detecting human in a crowd[J]. arXiv preprint arXiv:1805.00123, 2018.

[64] Zhang S, Benenson R, Schiele B. Citypersons: a diverse dataset for pedestrian detection[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017: 3213-3221.

[65] Ess A, Leibe B, Schindler K, et al. A mobile vision system for robust multi-person tracking[C]. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008: 1-8.

[66] Kendall A, Gal Y, Cipolla R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018: 7482-7491.

[67] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016: 770-778.

[68] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015: 3431-3440.

[69] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017: 2117-2125.

[70] Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the clear mot metrics[J]. EURASIP Journal on Image and Video Processing, 2008, 2008: 1-10.

[71] Ristani E, Solera F, Zou R, et al. Performance measures and a data set for multi-target, multi-camera tracking[C]. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016: 17-35.

[72] Sun S J, Akhtar N, Song H S, et al. Deep affinity network for multiple object tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 43(1): 104-119.

[73] Pang B, Li Y, Zhang Y, et al. Tubetk: Adopting tubes to track multi-object in a one-step training model[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020: 6308-6318.

[74] Peng J, Wang C, Wan F, et al. Chained-tracker: chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking[C]. European Conference on Computer Vision (ECCV), Glasgow, US, 2020: 145-161.

[75] Zhou X, Koltun V, Krähenbühl P. Tracking objects as points[C]. European Conference on Computer Vision (ECCV), Glasgow, US, 2020: 474-490.

[76] Pang J, Qiu L, Li X, et al. Quasi-dense similarity learning for multiple object tracking[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021: 164-173.

[77] Xu Y, Ban Y, Delorme G, et al. Transcenter: transformers with dense queries for multiple-object tracking[J]. arXiv preprint arXiv:2103.15145, 2021.

[78] Yu E, Li Z, Han S, et al. Relationtrack: relation-aware multiple object tracking with decoupled representation[J]. arXiv preprint arXiv:2105.04322, 2021.

[79] Tokmakov P, Li J, Burgard W, et al. Learning to track with object permanence[C]. IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021: 10860-10869.

攻读硕士学位期间取得的成果

项目

[1] **罗文斌**，杨健翔. 基于大数据框架 Flink 的数字供应链异常预警系统.2021.04-2021.08

[2] **罗文斌**，吴迪. 基于微服务平台的服务发现异常预警系统.2020.12-2021.03

比赛

[3] **罗文斌**，李颖，何小芳. CV101 计算机视觉青年开发者榜单.复赛 24 名,2019.11

[4] **罗文斌**，蔡兆祥，张梦璐.“数字人体”视觉挑战赛·宫颈癌风险智能检测诊断.复赛 18 名,2019.12

[5] **罗文斌**，李智敏，艾宏峰. 2020 年全国水下机器人（湛江）大赛.光学图像组复赛 31 名，2020.4