北京科技大学
University of Science and Technology Beijing

密级：公开

本科生毕业设计（论文）

题 目：基于 ST-GCN 与关键点检测的室内游泳者溺水识别研究

作 者：袁浩天

学 号：41918055

学 院：高等工程师学院

专 业：自动化（卓越计划）

成 绩：

2023 年 05 月

本科生毕业设计(论文)

题　　目: 基于 ST-GCN 与关键点检测
室内游泳者溺水识别研究

英文题目: Drowning recognition of indoor swimmers based on ST-GCN and key point detection

学　　院: 高等工程师学院

班　　级: 自动化 E191

学　　生: 袁浩天

学　　号: 41918055

指导教师: 张卫冬 **职称:** 研究员

指导教师: 艾轶博 **职称:** 副研究员

声　明

本人郑重声明：所呈交的论文是本人在指导教师的指导下进行的研究工作及取得研究结果。论文在引用他人已经发表或撰写的研究成果时，已经作了明确的标识；除此之外，论文中不包括其他人已经发表或撰写的研究成果，均为独立完成。其他同志对本文所做的任何贡献均已在论文中做了明确的说明并表达了谢意。

学生签名：_____　2023 年 5 月 16 日

导师签名：_____　2023 年 5 月 16 日

毕 业 设 计（论 文）任 务 书

一、学生姓名：袁浩天　　学号：41918055

二、题目：基于 ST-GCN 与关键点检测的室内游泳者溺水识别研究

三、题目来源：真实 ☑ 、　　自拟 ☐

四、结业方式：设计 ☑ 、　　论文 ☐

1. 查阅相关文献，深入了解溺水时人体外部形态特征，学习基础编程方法（Pytorch、Tensorflow），学习人体姿态估计算法、行为识别网络结构；

2. 选取合适的目标跟组算法、姿态估计算法和行为识别网络；

3. 收集相关数据集，并对数据进行预处理；

4. 构建适当的深度学习模型，并进行模型训练；

5. 使用测试集对模型进行检测，根据检测准确度修改优化模型；

六、主要（技术）要求：

1.学习目标检测算法、姿态估计算法和人体行为识别算法；

2.学习 Python 建模语言，能够运用 Python 对目标进行辅助建模；

3.学习 R-CNN 等深度学习网络框架。

4.选取合适的检测模型对数据集进行检测。

5.危险行为识别算法的检测率应大于等于 85%。

七、日程安排：

第七学期：

第 17-18 周：查阅资料，学习溺水人体检测理论，了解人体姿态估计的相关背景、基本原理以及研究现状，熟悉行为识别网络；

第八学期：

第 1-3 周：完成开题报告撰写，明确项目的主要内容和主要技术要求。通过查阅文献和书籍资料，了解课题背景以及国内外相关的研究现状。；

第 4-6 周：使用 Python 语言建立检测模型，根据数据集进行训练；

第 7-8 周：完成初步仿真试验，完成中期检查；

第 9-11 周：与实际场景比对，建立数字孪生模型，并进行仿真验证与模型优化；

第 12-14 周：开始撰写论文；

第 15-16 周：制作答辩 PPT，准备答辩。

八、主要参考文献和书目：

[1] 朱林.基于改进背景差分法的水下人体检测技术研究[D]. 北京: 北京工业大学,2017.

[2] 雷飞, 朱恒宇, 欧家豪, 王蕊, 张轩. 一种基于 YOLOv4 的泳池溺水检测方法. CN Patent 202110488324.4. Jul 23, 2021.

[3] 乔羽. 基于 Mask R-CNN 泳池中溺水行为检测系统的设计与实现[D]. 青岛大学, 2019.

[4] 邹旭, 廖钟豪, 王廷军, 等. 基于 ZigBee 通信模块的泳池防溺水智能泳帽的研究[J]. 科技风, 2018.

[5] 孙晓红. 2224 铝合金板材的疲劳性能研究[D].中南大学,2014.

[6] 胡裕超,杨辉.基于 ANSYS Workbench 的轮毂弯曲疲劳分析[J].汽车实用技术,2021,46(12):90-92.

[7] 霍梅梅,蔡建平,吴剑钟.基于运动分量阈值和机器学习的溺水检测方法和系统: 中国, CN110793539A[P]. 2020-02-14.

[8] 周海赟,项学智,翟明亮,等.结合注意力机制的深度学习光流网络[J].计算机科学与探索, 2020, 14(146):124-133.

[9] 薛丽霞, 江迪, 汪荣贵, 等. 融合注意力机制和语义关联性的多标签图像分类[J]. 光电工程, 2019, 46(09):22-30.

[10] 彭婷, 沈精虎, 乔羽. 基于改进 Mask R-CNN 的泳池溺水行为检测系统设计[J]. 传感器与微系统, 2021, 40(01):94-97.

[11] Stauffer C, Grimson W E L. Learning patterns of activity using real-time tracking[J]. IEEE

[12] Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8):747-757.

[13] Salehi N, Keyvanara M, Monadjemmi S A. An automatic video-based drowning detection system for swimming pools using active contours[J]. Int. J. Image, Graph. Signal Process, 2016, 8(8): 1-8.

[14] Dulhare U N, Ali M H. Underwater human detection using Faster R-CNN with data augmentation[J]. Materials Today: Proceedings, 2021.

[15] Adam George K.. Real-Time Performance and Response Latency Measurements of Linux Kernels on Single-Board Computers[J]. Computers, 2021, 10(5).
[16] Dulhare U N, Ali M H. Underwater human detection using Faster R-CNN with data augmentation[J]. Materials Today: Proceedings, 2021.

指导教师签字：　　2023 年 1 月 5 日

学 生 签 字：　　2023 年 1 月 5 日

系（所）负责人章：　　年　　月　　日

摘 要

随着社会经济的不断发展，人们的物质条件不断丰富，生活水平不断提高，人们更加注重身体健康，越来越多的人选择在游泳馆游泳健身。但是，游泳健身在帮助人们强身健体的同时也带来了诸多安全隐患。即使室内游泳馆基本上都有专业的救生人员，但是由于人眼检测会出现视觉疲劳等状况，并且在游泳人员拥挤或者救生人员过少的时候，极其容易出现救生人员不能及时发现水下游泳者的异常情况。因此，需要采用视频检测的方式，更灵敏高效地进行溺水检测。本文采用目标检测与跟踪、关键点检测与行为识别的方法，对室内游泳馆游泳者进行溺水检测，主要内容如下：

（1）本文使用基于 YOLOv3 的改进算法 PP-YOLO 进行目标检测。采用水下视频监测的方式，因此视频的分辨率及清晰度会有一定程度的影响。所以直接采用 YOLOv3 的检测效果不好，通过替换 YOLOv3 的检测模型，修改相关参数，得到一个识别效果良好的 PP-YOLO 检测模型。

（2）本文使用百度飞桨 AI Studio 平台的 PaddleVideo 架构对数据集进行目标检测及关键点预推理，得到训练文件及标注文件。

（3）根据训练文件及标注文件训练基于 ST-GCN 的行为识别模型，得到训练效果良好的 ST-GCN 模型及模型权重，将此训练好的行为识别模型及模型权重导入到 PaddleDetection 架构中，实现了对视频内单目标的溺水识别。

（4）本论文研究得到了基于 ST-GCN 和关键点检测的单目标溺水检测模型，在以后可以继续进行多目标溺水检测。

关键词： 目标检测，关键点检测，时空图卷积神经网络，溺水检测

Drowning recognition of indoor swimmers based on ST-GCN and key point detection

Abstract

With the continuous development of social economy, people's material conditions are constantly enriched, the living standard is constantly improved, people pay more attention to physical health, more and more people choose to swim in the natatorium fitness. However, swimming fitness not only helps people keep fit, but also brings many security risks. Even if the indoor swimming pool basically has professional lifesaving personnel, but due to eye detection will appear visual fatigue and other conditions, and in the crowded swimmers or lifesaving personnel is too few, it is extremely easy to appear lifesaving personnel can not find underwater swimmers timely abnormal situation. Therefore, video detection is needed to detect drowning more sensitively and efficiently. In this paper, the methods of target detection and tracking, key point detection and behavior recognition are adopted to detect drowning of swimmers in indoor natatorium. The main contents are as follows:

(1) This paper uses PP-YOLO, an improved algorithm based on YOLOv3, for target detection. The underwater video monitoring method is adopted, so the resolution and clarity of the video will be affected to some extent. Therefore, the detection effect of directly using YOLOv3 is not good. By replacing the detection model of YOLOv3 and modifying relevant parameters, a PP-YOLO detection model with good recognition effect is obtained.

(2) PaddleVideo architecture of Baidu Feizuo AI Studio platform was used in this paper for target detection and key point pre-reasoning of data set, and training files and annotation files were obtained.

(3) The behavior recognition model based on ST-GCN was trained according to the training files and annotation files, and the ST-GCN model and the model weight with good training effect were obtained. The trained behavior recognition model and the model weight were imported into the PaddleDetection framework to realize the drowning recognition of single object in the video.

(4) In this paper, a single target drowning detection model based on ST-GCN and key point detection is obtained, which can continue to carry out multi-target drowning detection in the future.

Key Words： Target Detection， Key Point Detection， ST-GCN， Drowning detection

目 录

1 引 言

随着社会经济的不断发展，人们的物质条件不断丰富，生活水平不断提高，现在的人们更加注重身体健康。目前，游泳作为一项健身运动，不仅能锻炼人们的心肺能力，还能帮助人们保持良好的体型。在游泳馆游泳健身，越来越受到人们的欢迎，更多的儿童、年轻人以及老年人开始投入到游泳健身的浪潮。但是，游泳健身在帮助人们强身健体的同时也带来了诸多安全隐患。不管是游泳初学者还是游泳运动员，在游泳的时候，都可能面临溺水的危险状况。根据世界卫生组织统计，全世界每年有超过 37 万人因溺水而死[1]。根据调查显示，在我国一年大约有 60000 人死于溺水，这意味着每天超过 150 死于溺水。所以，在室内游泳池中进行游泳状况的监测，有着无可替代的研究意义。

为了尽量避免室内游泳馆出现游泳者溺水身亡的情况，现在的室内游泳馆基本上都有专业的救生人员，来监测游泳者是否出现异常行为。但是由于人工会存在疲劳等状况，并且在游泳人员拥挤或者救生人员过少的时候，极其容易出现救生人员不能及时发现水下游泳者的异常情况。因此，仅靠数量有限的救生人员，无法真正有效地识别水下游泳者的异常情况，避免出现游泳者溺水身亡的事故。

近年来，随着计算机视觉和深度学习网络的迅速进步，人体行为识别逐渐成为了热门领域之一。通过利用计算机视觉和图像处理技术，将实时监控区域内人的图像和视频流进行分析和处理，识别出人体行为并进行一系列操作，可以在各种各样的领域及生活中得到丰富的运用。通过将人体行为识别应用到溺水检测当中，既可以避免人工检测可能出现的因疲劳导致的漏检，又可以更加及时地将游泳者的溺水行为进行反馈、报警，方便及时展开救助。同时，通过搭配在水下进行视频监测、溺水识别，能更加有效地避免因水面反光或是人影重叠导致的溺水检测不及时，可以更加有效地保证室内游泳者的人身安全。因此，建立基于人体行为识别的室内游泳者溺水检测，具有很大的意义。

本文着眼于室内游泳馆水下监测视频，立足于单目标识别与跟踪技术，提出一种基于 ST-GCN 和关键点检测的室内游泳者溺水识别方案。

本文的组织安排结构如下：

第一章，引言。

本章讨论本篇论文的研究背景、研究目的与研究意义。

第二章，国内外研究现状。

本章主要概述目标检测的研究现状，人体关键点检测与姿态估计算法、基于关键点检测的行为识别算法以及溺水检测的研究历程以及现状。

第三章，目标检测相关知识介绍。

本章介绍目标检测相关知识，先介绍 YOLOv3 网络结构，介绍 YOLOv3 的网络输入、网络结构以及目标检测过程；再介绍基于 YOLOv3 的 PP-YOLO 模型，介绍 PP-YOLO 网络的检测原理、检测流程以及需要修改的参数。

第四章，关键点检测与行为识别相关知识介绍。

本章介绍基于时空图卷积神经网络的行为识别算法，包括关键点检测模型 HRNet 和时空图卷积神经网络的网络结构、相关介绍以及 ST-GCN 的实现。

第五章，基于 ST-GCN 与关键点检测的溺水检测。

本章基于 ST-GCN 与关键点检测，借助百度飞桨 AI Studio 平台，实现对室内游泳者的溺水识别。包括数据采集与预处理、人体关键点检测与姿态估计、模型训练、模型测试以及模型检测结果可视化。

第六章，总结与展望。

本章主要回顾整篇论文所述，对得到的结论加以展现，同时列举本文的不足之处以及需要改进的地方，对未来的研究方向进行展望。本文技术路线图如图 1-1 所示：

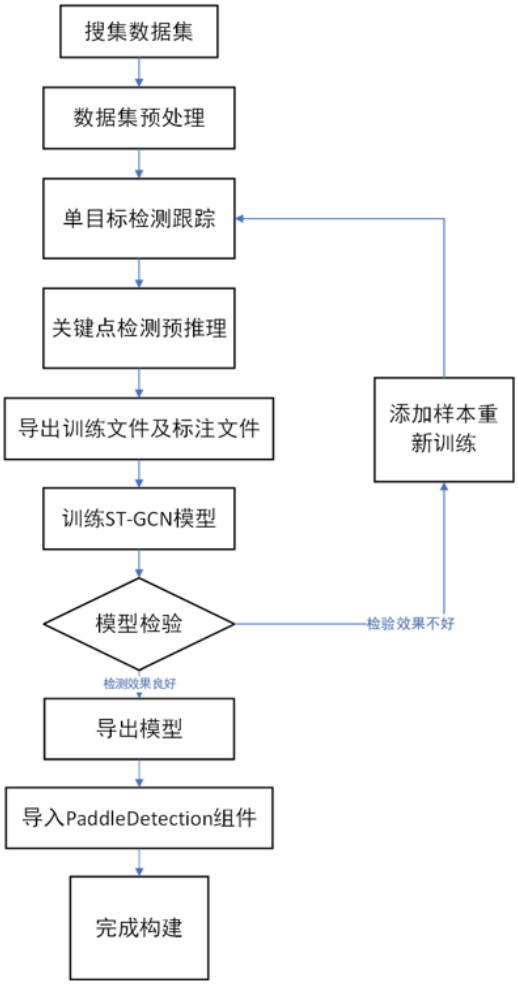
搜集数据集
数据集预处理
单目标检测跟踪
关键点检测预推理
导出训练文件及标注文件
训练ST-GCN模型
模型检验
检验效果不好
检测效果良好
导出模型
导入PaddleDetection组件
完成构建
添加样本重新训练

图 1-1 技术路线图

2 国内外研究现状

本文研究基于 ST-GCN 与关键点检测的室内游泳者溺水识别，主要涉及目标检测与跟踪、人体关键点检测和行为识别领域的知识，下面对相关领域的研究现状做文献调研。

1975 年，瑞典乌普萨拉大学的 Johansson[2]在实验中发现，由一系列关键点的位置信息进行组合，可以对特定行为进行描述。同时并以这种理念为基础，建立了由十二个关键点组成的人体模型。由此，人体的各个行为动作特征信息，就可以由这些关键点的位置反映出来。这为当时的人体行为识别提供了一种崭新的思路。随后随着技术不断进步，关于人体行为识别的技术迄今为止大致经历了三个阶段：从 1975 年到 1990 年，人体行为识别技术领域主要关注的问题是该如何建立分析人体行为的识别模型；从 1990 年到 2010 年，主要研究的方向是根据一些简单的人体行为动作，识别出人体的行为信息，如手势识别等；从 2010 年到现在，计算机硬件水平不断提高，深度学习网络也不断发展成熟，建立起来的人体行为识别算法也依托于技术的进步在根据图片、视频进行人体行为识别方向取得了很大突破。

2.1 目标检测研究现状

目标检测技术是计算机视觉领域中的一个重要的研究方向，在早期的研究历程中，该算法主要采用基于手工特征的方法，如边缘、角点等，以及基于滑动窗口和分类器的方法，如 HOG+SVM 等[3]。这些方法在一些简单场景下具有较好的表现，但在复杂背景下的性能表现欠佳。除此之外，这些方法不仅步骤繁琐，窗口冗余问题也很严重。2012 年 AlexNet[4]的提出，意味着在计算机视觉领域，深度学习技术也开始得到了较为广泛的应用实践。越来越多的科研项目开始借助深度学习技术，实现更加精确高效的计算机视觉检测。比较明显的例子就是，在目标检测领域，一些基于神经网络模型的目标检测算法开始被研究出来，如 R-CNN、Fast R-CNN、Faster R-CNN、YOLO 等。这些方法利用深度卷积神经网络对图像进行特征提取和目标定位，在此基础上，这些基于深度卷积神经网络的目标检测算法在训练和推断的过程中也实现了端到端的更新。基于深度卷积神经网络的目标检测技术的种种改进方案，使得这些新型目标检测算法具有更精确的检测效果，也有更快速的检测速度。但是由于深度学习技术的发展还不够彻底完善，深度学习的一些缺点，例如实时性差等特点也使得基于深度学习的目标检测算法仍然

存在检测实时性差、检测精度低的问题，除此之外，目标检测技术还在应用中遇到了一些瓶颈，如检测精度和速度之间的平衡，对小目标的识别能力，对多尺度目标的识别等。针对这些问题，出现了很多细化和优化的算法，如SSD、RetinaNet、Mask R-CNN 等。同时，也有一些新的方向被提出来，如one-stage detector[5]、anchor-free detector[6]等。

目前，目标检测技术已经逐渐成熟并被广泛应用于各个领域，例如视频监控、自动驾驶、机器人等。现阶段，由于要考虑精度和速度的平衡，一些基于 RetinaNet[7]和 EfficientDet[8]的方法已经能够在保证较高检测精度的情况下实现比较快的检测速度。针对多目标检测目标之间的相互遮挡和干扰的问题，一些基于 Mask R-CNN 和 Detectron2 的方法已经能够实现较好的多目标检测表现[9][10][11]；同时，在小目标检测领域，由于小目标检测涉及到目标尺寸和背景复杂性等多种因素的影响，对检测器的鲁棒性和精度提出了很高要求。目前，一些基于 SSD 和 YOLOv4 等方法已经能够实现较好的小目标检测表现[12][13][14]；随着移动设备的普及和应用场景的多样化，轻量化的目标检测模型已成为一种研究热点。目前，一些基于 MobileNet、ShuffleNet 和 EfficientNet 等的轻量级目标检测模型已经被提出[15][16][17]，可以在资源受限的情况下实现目标检测任务；对于弱监督目标检测，弱监督目标检测是指基于图像级别标签，通过利用弱监督学习算法和目标分割技术，进行目标检测的一种方法。该方法不需要对图像的每个目标都进行像素级别的标注，降低了数据标注的成本，但是其检测精度有待进一步提高[18][19]。

综上所述，目标检测技术的研究正在不断深入，涉及到多个方面，包括模型设计、数据增广、损失函数的优化等等。未来，随着技术的不断进步和应用场景的不断扩展，目标检测技术必将得到进一步的提升。

2.2 基于关键点检测的行为识别算法研究现状

2.2.1 关键点检测与姿态估计

人体关键点检测与姿态估计技术被广泛应用于弱监督学习、人机交互、虚拟现实、监控安防等领域[20][21]。例如，关键点检测可以用于人体动作识别、手势识别、头部姿态估计等任务。针对人体的行为识别技术主要是依赖于根据人体各种行为进行建模的方法，而人体在进行各种动作行为时，身体的大部分会展现出柔韧性、伸缩性，而对于人体的关节而言，在人体进行各种行为活动时，其作为一种人体的“关键点”并不会展现出柔韧性和伸缩性，而是会保持稳固的特点，所以各个关节之间连接的人体段落是稳固的。

同时，在面临众多外部条件，如衣物遮挡、光照、能见度低等，人体在这些外部条件下，进行各种行为活动，其人体的关键点就成为了可以提供最有效稳定的人体数据来源。由此可见，人体的关键点数据是建立针对人体的行为识别模型的稳定数据来源以及重要的姿态估计基础。在进行人体姿态估计时，有 2D 姿态估计和 3D 姿态估计两种方法，他们分别检测到人体的 2D 关键点和 3D 关键点。在进行基于 2D 关键点的姿态估计时，首先要为每一个 2D 关键点预测一个二维坐标(X,Y)；在进行基于 3D 关键点的姿态估计时，为每一个 3D 关键点预测一个三维坐标(X,Y,Z)。由于这些 2D 关键点和 3D 关键点在世界上给人带来了明显的差异，因此人体的关键点检测与姿态估计非常具有挑战性。人体姿态估计，主要是为了对人体的各个关键点实现检测，然后根据这些关键点重建肢体。目前实现人体姿态估计的思路方法主要有两种：一是自下而上的方式，先对图像或者视频中的人体进行关键点检测，再根据关键点得到预测的肢体从而得到人体目标姿态；二是自上而下的方式，先检测得到人体目标，再通过关键点检测得到一堆关键点数据，然后根据关键点预测得到肢体，进而得到人体目标姿态。

近年来，随着深度学习技术的不断发展，在人体关键点检测领域也常常见到深度学习技术的广泛应用。常用的深度学习模型包括 Hourglass、SPPE、HRNet 等[22][23][24]，这些模型通过卷积神经网络进行特征提取和特征优化，得到了比传统方法进行关键点检测更精确更高效的检测结果，可以在较高的精度下准确检测出人体关键点。

深度学习算法需要依赖大规模的数据集进行训练。在人体关键点检测与姿态估计领域，有很多著名的常用数据集：COCO 数据集，微软公司开发的数据集，包含超过 20 万张含有人类实例的图片。其中使用了超过 20 万个实人标注数据，使得该数据集在人体关键点检测任务中具有很高的准确性；MPII Human Pose 数据集，这是由德国马克斯普朗克研究所开发的数据集，包含跨越多种活动类型的 25,000 张图像。这些图像中包含公共场合人物的姿态，包括跑步、骑自行车、游泳等；3DHumanPose 数据集，华盛顿大学开发的数据集，通过摄像机捕捉技术收集了大量人类运动数据，包括跳舞、健身等。该数据集采用了 3D 模型技术，可以提高姿态估计的准确度。

在算法方面，人体关键点检测与姿态估计也有很多比较常用的算法：Faster R-CNN[25]：这是经典的目标检测算法，通过在 Fast R-CNN 的基础上引入 RPN 网络进行候选框生成，成功应用于人体姿态估计中；Mask R-CNN[26]：是 Faster R-CNN 的改良版，可实现目标检测像素级别的分割。在人体关键点检测任务中，可以通过 Mask R-CNN 准确地分割出每个人的轮

廓，然后再进行关键点的识别；OpenPose[27]：这是一种新的人体姿态估计框架，通过图像解析技术和 CNN 网络结合，可以实现在人体关键点检测和姿态估计技术中取得更精确更高效的算法结果。对于多人姿态估计，是指针对多个人体同时进行关键点检测的问题。由于存在相互遮挡和姿态变化等因素，多人姿态估计具有较高的难度。目前，一些基于 Top-down 和 Bottom-up 的多人姿态估计方法已经能够实现精确的关键点检测。除此之外，一些基于轻量级模型和 GPU 加速等技术的实时人体关键点检测方法已经被提出，如 MobileNet、EfficientNet 和 Yolact 等[28][29][30]。

2.2.2 关键点检测与行为识别

基于关键点的行为识别在计算机视觉领域一直是很大的一个研究热点，因为它在许多现实世界的应用中发挥着重要作用，例如智能监控，人机交互等。它旨在使用人体关键点识别人类行为，并在具有复杂背景，例如杂乱场景，光线条件的动态环境中显示出优势。

针对人体行为识别，一般可以采用时间序列建模方法，如 HMM、LSTM 和 GRU 等循环神经网络[31][32][33]，或者注意力机制方法，如 Self-Attention、Transformer 等[34][35]。这些方法在时间序列处理和特征融合上都有不错的表现。在进行人体行为识别时，需要注意两个重要特性：一是实时性，实时性是人体行为识别技术的一个重要指标，特别是在一些需要快速反应的应用场景中，如智能监控等。一些基于轻量级模型、优化算法和硬件加速等技术的实时行为识别方法已经被提出，如 Real-time Action Recognition (RAR)[36]、Real-time Multi-person Pose Estimation and Tracking (PoseFlow)[37]等。二是可解释性，可解释性是指算法能够对结果做出合理解释，使人能够理解算法的决策过程，这在应用场景中尤其重要。一些基于可视化和解释性技术的行为识别方法已经被提出，如多尺度可视化实例分割模型（MS-DI）等[38]。

但是，人体行为识别仍存在识别准确率较低的问题，针对这个问题，很多研究人员尝试采用注意力机制来提高模型的检测准确度。Yan[39]等人采用卷积分层注意力模型，构建 LSTM 的分层系统来对视频中的人体进行目标行为识别。Sharma[40]等人采用对一段视频中关键帧的人体各个部位进行选择性的识别，结合视觉注意力模型，构建出新的 LSTM 模型，该模型在基于 3D 关键点信息的人体行为识别中具有良好的性能。Yan[41]等人又提出的 ST-GCN 模型，是一种空时图卷积网络模型，该模型与其他模型最大的区别，也是最突出的优势就是建立了动态的人体关键点模型，通过将检测出来

的人体关键点结合空时图卷积神经网络，大大提高了人体行为识别的检测精度和检测效率。

2.3 溺水检测研究现状

相比于国内溺水检测的刚起步阶段，国外在四十年前便开始在溺水检测领域展开相关研究。针对检测方式，室内游泳馆的溺水检测可分为硬件检测和视频检测。硬件检测大部分由人体佩戴穿戴式设备，这种检测方式相对安全可靠，但是为游泳者带来了不便；视频检测可分为水下视频检测和水上视频检测，这种检测方式最大的优势就是方便游泳者，且检测迅速，但是也会存在水下游泳者肢体遮挡，水上视频水面反光等问题。

瑞士的 Blue Fox[42]于 2008 年被发明，它是一种穿戴式的溺水检测设备，由佩戴在游泳者腕部的微型电脑，与安装在游泳池池壁的接收装置相配合。游泳者佩戴的微型电脑时刻检测游泳者所处的水深以及所处水深的时间，当游泳者所处水深超过设定值时，以及在一段水深所处时间过长时，微型电脑会向接收装置发送信号使其发出报警信号。以色列的 coral manta[43]是一种基于水下检测视频的溺水时别系统，它时刻检测游泳池内游泳者所处的水深，如果检测到游泳者长时间位于水下某深度时，就会自动反馈到手机应用程序里报警。这些系统相对安全，但是也会存在易于误报的情况，并且可能会出现反馈不及时导致溺水事故出现的情况。

2019 年厦门闻达科技有限公司发明的可穿戴式溺水报警器，将无线安全信号作为溺水判断条件，极大地减少了溺水检测的误报率，下图为该溺水检测装置的检测流程图。

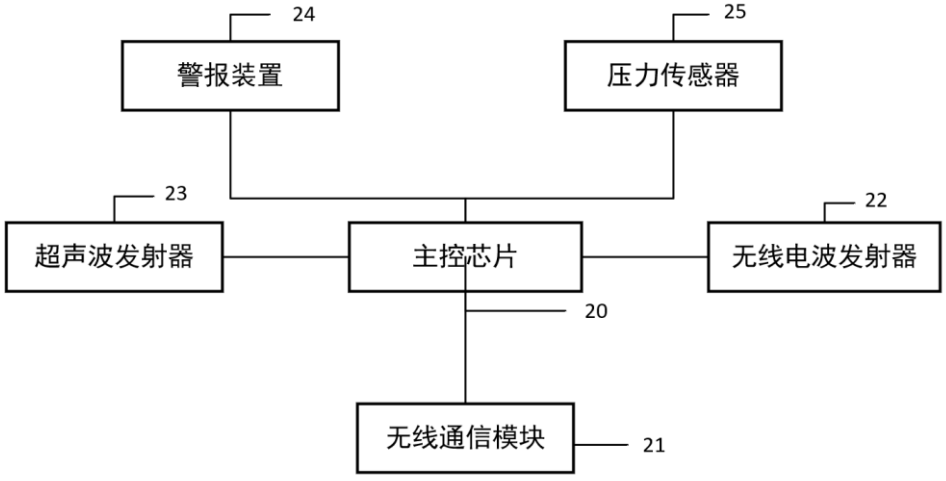

图 2.1 溺水检测流程图

2016 年，吴婷璇[44]提出一种基于水上视频监测的室内游泳者溺水检测与跟踪算法，该篇论文首先对水上视频进行降噪处理，较弱了水面波纹对于溺水检测的影响，然后比较几种不同的背景差分法以及几种目标检测算法，取效果最好的算法给出状态转移图。2019 年，乔羽[45]等人提出了一种基于 Mask R-CNN 的溺水检测系统，这是一种基于视频图像的溺水检测方法。

综上所述，目标检测算法有 YOLO 系列，以及在此技术上改进的例如 PP-YOLO 等算法，以及 SSD、RetinaNet、Mask R-CNN 等算法。人体关键点检测模型有 Hourglass、SPPE、HRNet 等，行为识别模型有 HMM、LSTM、ST-GCN 和 GRU 等。可以将其组合起来，训练得到本文研究的室内游泳者溺水行为识别模型。

3 目标检测相关知识介绍

3.1 YOLOv3 网络

YOLOv3 凭借其较之于前两代较大的结构变化以及强大的性能，现如今在各个领域仍然被广泛使用。YOLOv3 于 2018 年被提出，采用的是残差结构，并在预测阶段采用多层特征金字塔的网络结构[54]。相较于前两代，YOLOv3 具有更精确的检测精度，并且它的检测速度也十分迅速，几乎可以达到实时监测的效果。整个 YOLOv3 的模型结构如下图所示。

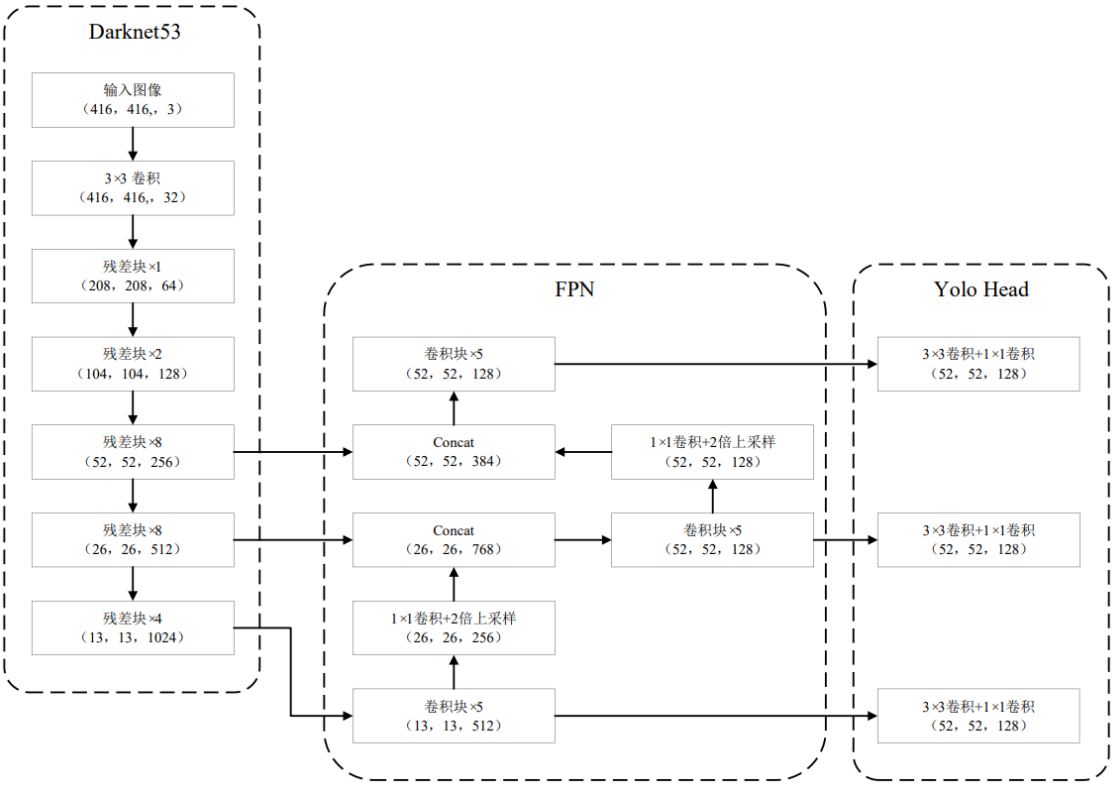

图 3.1 YOLOv3 模型结构

如上图所示，YOLOv3 一次次地将特征图的宽和高进行压缩，同时扩张通道数，从而对输入的图片进行多级的特征提取，然后通过上采样的方式，对这些多级特征提取的特征进行采样，从而获得一系列特征层，最后传入特征金字塔结构中，以此重复进行三次，将得到的三种不同的特征层进行堆叠，用于最终的目标检测。

3.1.1 网络输入

由图 3.1 可知，在 YOLOv3 网络机构中，至少会存在五次采样，每次采样的步长为 2，因此网络的最大步幅为 2^5=32，所以网络输入的图片大小必须为 32 的整数倍。

3.1.2 DarkNet-53

YOLOv3 采用了 Darknet-53 作为特征提取网络。Darknet-53 是一个包含 53 层卷积神经网络的模型，它可以从原始图像中提取出高层次的语义特征，如形状、纹理和颜色等。与其他流行的卷积神经网络相比，例如 ResNet 和 Inception 等，Darknet-53 更加轻量级，同时具有较好的性能。如图 3.1 所示，它由 Darknet-53 的主要结构包括多个卷积层、批量归一化层、线性整流层和残差块。其中，残差块是 Darknet-53 的核心模块，它可以有效地解决深度神经网络中的梯度消失和梯度爆炸问题[46]。

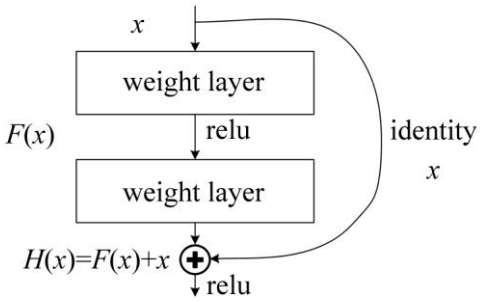

图 3.2 残差块结构图

所加入的残差块结构图如图 3.2 所示。在 Darknet-53 中，每个残差块包含两个卷积层和一个残差连接。在残差连接中，输入特征图被直接添加到残差块的输出特征图中，从而使得模型可以更加深层次地进行特征抽取和处理。残差块的计算公式如式(3-1)所示。

$$W_{EMA} = \lambda W_{EMA} + (1-\lambda)W \tag{3-1}$$

此外，Darknet-53 还采用了高分辨率上采样和低分辨率下采样等技术来增加网络的感受野和提高特征提取的效果。具体来说，这些技术可以对输入图像的不同尺度和大小进行适应性处理，并更好地捕获目标物体的特征。

3.1.3 目标预测

YOLOv3 使用经典的逻辑回归方法来预测每个边界框的置信度。如果边界框与特征层的交并比大于规定的阈值，则认定该边界框的置信度为 1，表示在这个边界框内，所需检测的目标是存在的；如果边界框与特征层的交并比小于规定的阈值，则认定该边界框的置信度为 0，表示所需检测的目标是不存在的。

如图 3.1 中的 FPN 结构，是为了提高目标检测准确度所采用的特征金字塔结构。当获得一层特征层之后，与未来特征层进行上采样，从而将低层与高层的特征相融合，提高了检测准确度。

3.2 基于 YOLOv3 改进的 PP-YOLO 模型

PP-YOLO 是百度飞桨团队基于 YOLOv3 算法提出的性能更加优良的目标检测器，主要尝试结合现有的各种不增加模型参数和 FLOPs 数量的技巧，以达到在保证速度几乎不变的情况下尽可能提高检测器精度的目标。PP-YOLO 可以在有效性(45.2% mAP)和效率(72.9 FPS)之间实现更好的平衡，超越了现有一些先进的探测器，如 EfficientDet 和 YOLOv4。PP-YOLO 和其他先进目标检测器的检测速度比较如下图所示。由图可知，PP-YOLO 与其他先进的目标探测器的比较。PP-YOLO 比 YOLOv4 运行速度更快，mAP 从 43.5%提高到 45.2%。

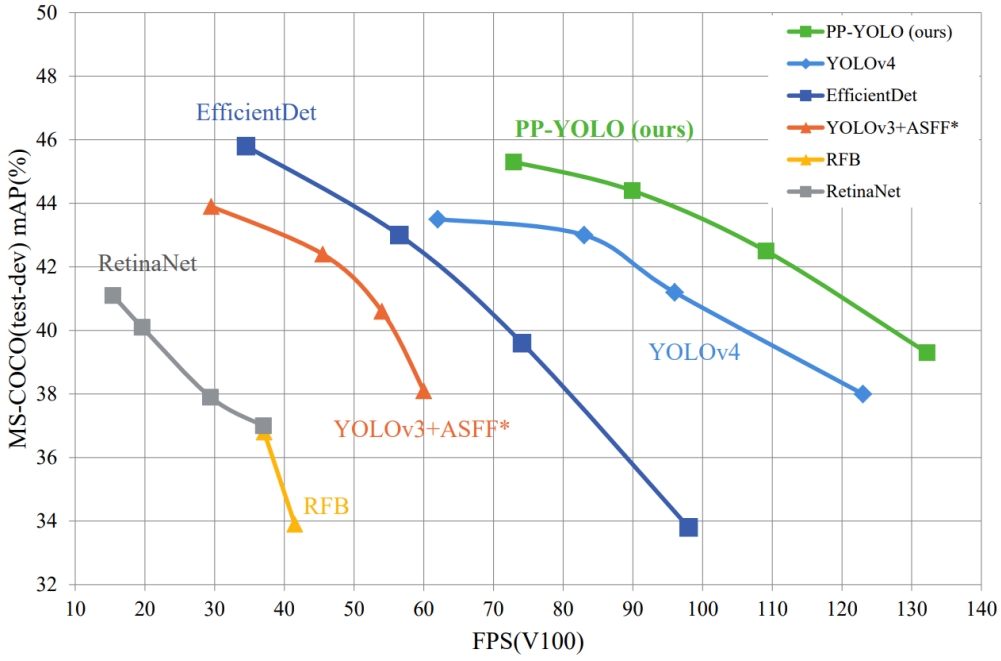

图 3.3 PP-YOLO 与其他先进目标检测器检测速度比较

3.2.1 PP-YOLO 检测原理

与 YOLOv3 采用的 Darknet-53 特征提取网络不同，PP-YOLO 使用了更加轻量级的 Backbone 网络中的 ResNet50-vd，因为 ResNet 得到了广泛的应用和更广泛的研究，可供选择的不同变体也更多，并且也通过深度学习框架得到了更好的优化。PP-YOLO 的网络结构如下图所示。

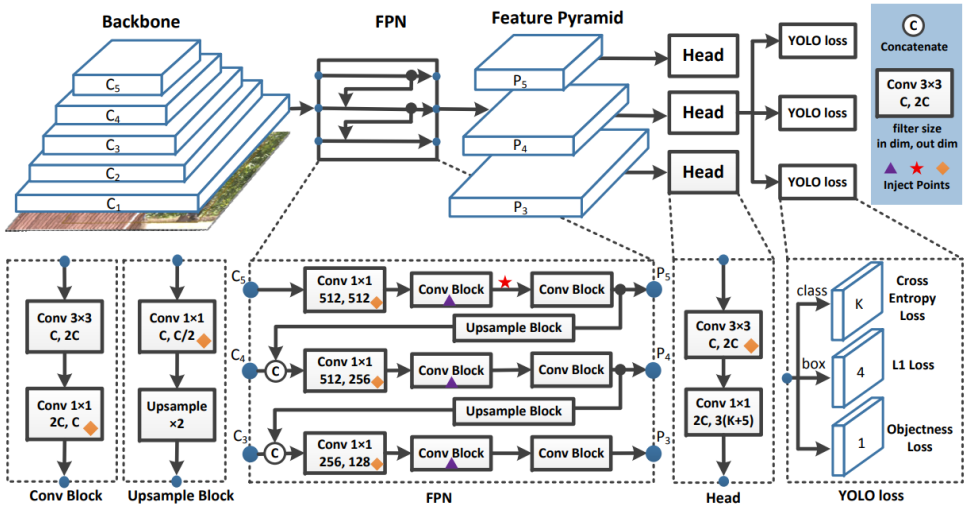

图 3.4 PP-YOLO 网络结构

考虑直接用 ResNet50-vd 替换 DarkNet-53 会损害 YOLOv3 探测器的性能。将 ResNet50-vd 中的一些卷积层替换为可变形的卷积层。变形卷积网络(Deformable convolutional Networks, DCN)的有效性已经在许多检测模型中得到验证。DCN 本身不会显著增加模型中的参数数量和 flop，但在实际应用中太多的 DCN 层将大大增加后处理时间。因此，为了平衡效率和效果，只在最后阶段用 DCN 替换 3 个×3 卷积层。我们将这个修改后的主干表示为 ResNet50-vd-dcn，第 3、4、5 阶段的输出表示为 C3、C4、C5，然后使用 FPN 构建特征映射之间具有横向连接的特征金字塔。特征图 C3、C4、C5 被输入到 FPN 模块。

3.2.2 PP-YOLO 检测流程

PP-YOLO 的检测步骤如下：

（1）首先，将待检测的图像进行预处理，包括图像大小的缩放、像素值的归一化等操作，以使其能够适应 PP-YOLO 模型输入的要求。这里输入

图片的要求与 YOLOv3 所输入的图片要求一致，即网络输入的图片大小必须为 32 的整数倍。

（2）使用特征提取网络 ResNet50-vd，对经过预处理的图像进行多层特征提取，得到图像的一系列特征图。然后，利用特征金字塔算法将不同尺度的特征图融合，以便于检测不同大小的物体。

（3）针对不同特征尺度，使用聚类以及 K-Means 算法，基于训练数据集的目标边界框宽高比例将输入的网络图像划分成一系列网格，生成一组默认锚框（anchor），由此产生候选区域。

（4）对于每一个特征图上的每个锚框，通过卷积神经网络预测该锚框中包含的物体的类别概率和边界框回归信息。

（5）利用非极大值抑制（NMS）算法，过滤重复的检测结果，仅保留检测质量最高的结果。同时，根据不同目标的置信度范围，对检测框进行颜色编码，以实现更好的可视化效果。

（6）将最终的检测结果（类别、位置、置信度）输出，并可通过图像绘制等方式呈现给用户。

PP-YOLO 的检测流程图如下图所示。

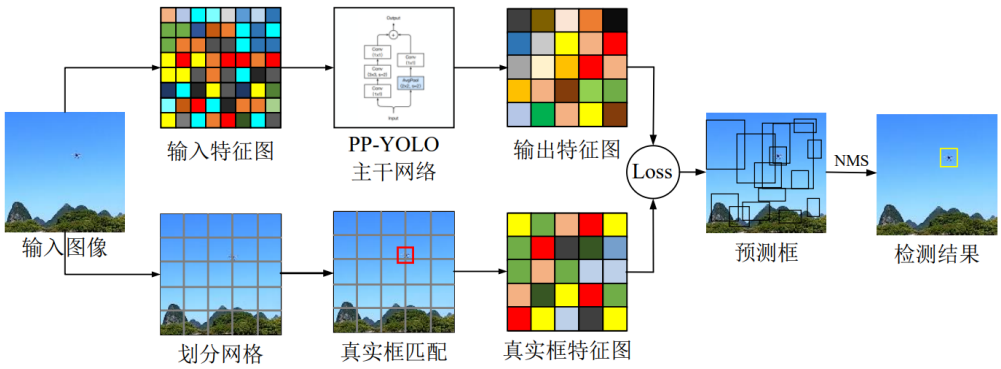

图 3.5 PP-YOLO 检测流程图

3.2.3 PP-YOLO 参数优化

PP-YOLO 采用了许多新颖的方法，它只是在 YOLOv3 上应用这些方法，并不是提出一种新的检测方法，所以这些方法很多不能直接应用在 YOLOv3 上，需要根据它的结构进行调整。

（1）批处理大小（batch size）

使用更大的 Batch Size 可以提高训练的稳定性，得到更好的结果。这里我们将训练批处理大小从 64 个更改为 192 个，并相应地调整训练时间表和

学习率。

（2）指数移动平均线（EMA）

在训练模型时，保持训练参数的移动平均线通常是有益的。使用平均参数的评估有时会产生明显优于最终训练值的结果。指数移动平均线(EMA)使用指数衰减计算训练参数的移动平均线。对于每个参数 W，我们维持一个阴影参数:

$$W_{EMA} = \lambda W_{EMA} + (1-\lambda)W \tag{3-2}$$

其中 λ 为衰减。我们采用衰减 λ 为 0.9998 的 EMA，并使用阴影参数 W_{EMA} 进行评价。

（3）正则化方法（DropBlock）

DropBlock 是结构化 dropout 的一种形式，其中特征图中相邻区域中的单元被放到一起。与原始方法不同的是，我们只将 DropBlock 应用于 FPN，避免将 DropBlock 添加到骨干导致性能下降。

（4）交并比损失（IoU Loss）

边界盒回归是目标检测的关键步骤。在 YOLOv3 中，边界盒回归采用 L1 loss。它不是为 mAP 评估指标量身定制的，mAP 评估指标强烈依赖于 Intersection over Union (IoU)。为了解决这一问题，提出了 IoU loss 和其他变化，如 CIoU loss 和 GIoU loss 损失。与 YOLOv4 不同的是，没有直接将 L1 loss 替换为 IoU loss，而是增加了另一个分支来计算 IoU loss。发现各种 IoU loss 的改进是相似的，所以选择最基本的 IoU loss。

（5）网络敏感度（Grid Sensitive）

增加网络敏感度是 YOLOv4 引入的一种有效的技巧。当我们解码边界框中心 x 和 y 的坐标时，在原始的 YOLOv3 中，我们可以通过以下两个式子得到它们：

$$x = s \cdot (g_x + \sigma(p_x)) \tag{3-3}$$

$$y = s \cdot (g_y + \sigma(p_y)) \tag{3-4}$$

但是由于 x 和 y 不能完全等于 $s \cdot g_x$ 或 $s \cdot (g_x + 1)$，这使得很难预测刚刚位于网格边界上的边界框的中心。我们可以将方程改为:

$$x = s \cdot (g_x + \alpha \cdot \sigma(p_x) - (\alpha - 1)/2) \tag{3-5}$$

$$y = s \cdot (g_y + \alpha \cdot \sigma(p_y) - (\alpha - 1)/2) \tag{3-6}$$

这使得模型更容易预测精确位于网格边界上的边界框中心。

（6）矩阵化非极大值抑制（Matrix NMS）

Matrix NMS 从另一个角度，将其他检测分数作为它们重叠的单调递减函数进行衰减，以并行的方式来实现。因此，矩阵式 NMS 比传统 NMS 速度更快，不会带来任何效率的损失。

（7）坐标变换（CoordConv）

它的工作原理是通过使用额外的坐标通道给卷积访问自己的输入坐标。CoordConv 允许网络学习完全的平移不变性或不同程度的平移依赖性。考虑到它会在卷积层中增加两个输入通道，为了尽可能减少效率的损失，不改变骨干中的卷积层，只将 FPN 中的 1x1 卷积层和检测头中的第一个卷积层替换为 CoordConv。

（8）空间金字塔池化（Spatial Pyramid Pooling, SPP）

SPP 通过对不同尺度的池化窗口提取不同尺度的池化特征，从而对不同大小的输入保持固定的输出，有效增加特征的感受野。PP-YOLO 将其运用于 FPN 的第一层（C5）特征后，以获取更丰富的语义信息，增加对小目标的感知能力。

综上所述，PP-YOLO 算法中的参数优化主要包括学习率调整、正则化技术、数据增强、网络结构调整以及损失函数等方面，这些优化方法可以为模型的训练和性能提升提供有力的支持。

4 关键点检测与人体行为识别相关知识介绍

4.1 关键点检测模型 HRNet

HRNet 是一种高分辨率网络，全称为 High-Resolution Network。它是由中国科学院计算技术研究所提出的一种卷积神经网络，其主要特点是能够高效地融合多个分辨率特征图，从而在保持高分辨率信息的同时，又能兼顾不同尺度物体的检测和识别。

早期的人体关键点检测网络，例如 Hourglass[46]、CPN[47]、SimpleBaseline[48]和 DeeperCut[49]等，都是借助下采样的方式，一次次地构建多尺度特征，最后通过上采样保证必要的高分辨率输出。HRNet 则认为像素级任务地人体关键点检测需要高分辨率特征地提取、保持和表达。它通过并行地链接高分辨率到低分辨率地子网，使用重复的多尺度融合，始终维持高分辨率的表示，HRNet 的结构图如下图所示。

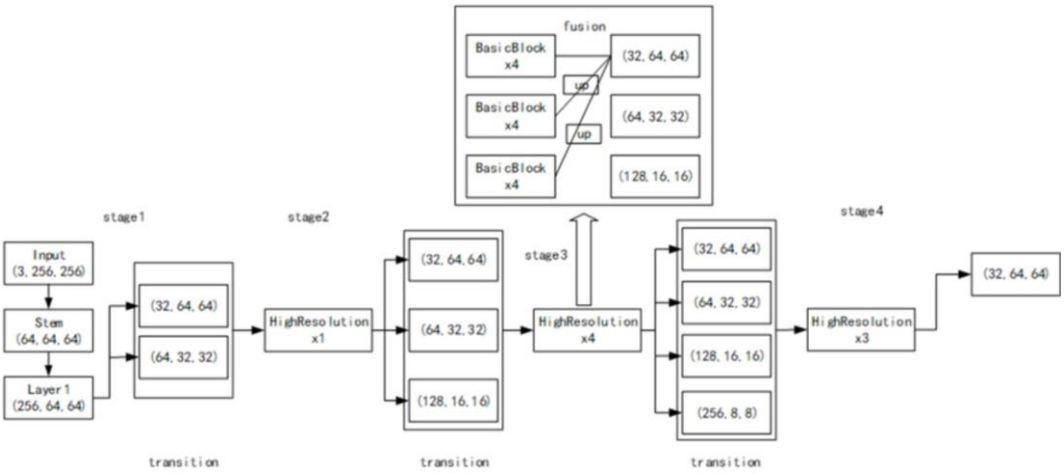

图 4.1 HRNet 结构图

HRNet 的网络结构非常简单，主要由四个阶段组成：高分辨率阶段、低分辨率阶段、高分辨率再现阶段和融合阶段[50]。其中，高分辨率阶段和低分辨率阶段用来提取不同分辨率的特征图，高分辨率再现阶段用于将低分辨率特征图上采样到和高分辨率特征图相同的尺寸，以保持高分辨率信息，最后的融合阶段将不同尺度的特征图融合起来作为网络的输出。融合方法如下

图所示：

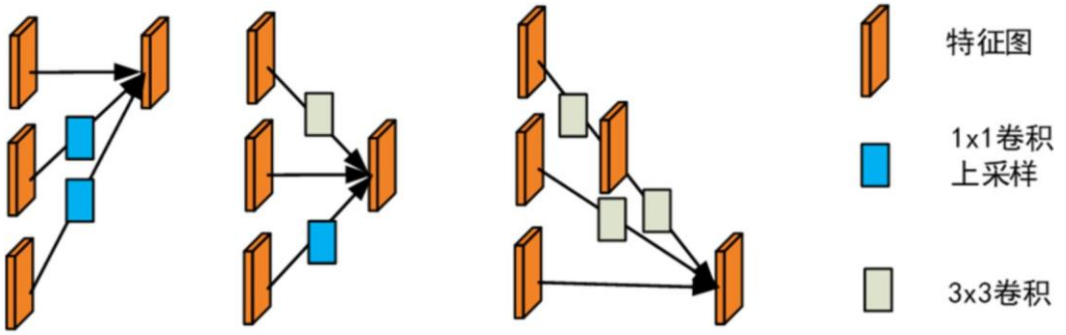

图 4.2 HRNet 融合过程

需要注意的是，HRNet 的分支网络中并没有池化操作，这是为了避免掉失原始的高分辨率信息[51]。同时，HRNet 的低分辨率特征图使用了多个分支来提取不同分辨率的特征，这也是为了在保持高分辨率信息的情况下，充分挖掘图像的多尺度信息。

4.2 时空图卷积神经网络 ST-GCN

4.2.1 网络结构

首先对视频进行姿态估计，对骨架序列构建时空图。将应用多层时空图卷积(ST-GCN)，并在图上逐步生成更高层次的特征图。然后通过标准的 Softmax 分类器将其分类到相应的动作类别。整个网络结构如下图所示。

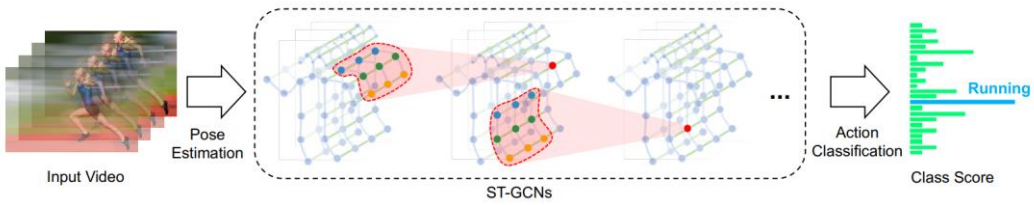

图 4.3 时空图卷积神经网络结构

4.2.2 图卷积神经网络

在我们深入研究完整的 ST-GCN 之前，我们首先看一下 GCN 模型。GCN 是指图卷积网络（Graph Convolutional Network），是一种用于处理图数据的深度学习模型。对于图数据，传统的神经网络方法无法直接处理，而 GCN 则可以通过图上的节点和边进行建模，从而能够对整个图结构进行学

习和预测。

与传统的卷积神经网络（CNN）不同，GCN 中的卷积操作并非使用局部滤波器来提取局部特征，而是利用邻接矩阵来计算每个节点的加权和，实现信息传递和特征聚合[52]。因此，GCN 可以同时考虑节点的属性和拓扑信息，从而得到更全面的表示。GCN 的网络构架如下图所示。

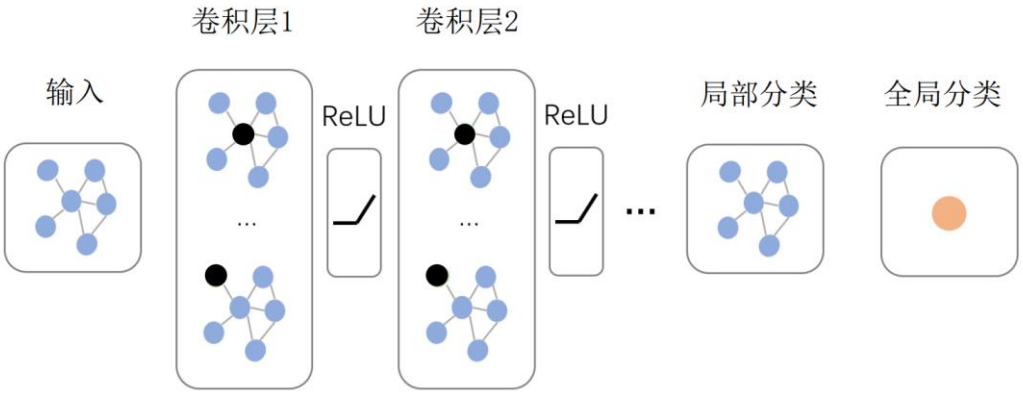

图 4.4 GCN 网络构架

由图 4.4 可知，将一个图片输入 GCN 网络之后，在第一个卷积层内，对每一个节点附近都进行一次卷积操作，再经过激活函数 ReLU，以此类推重复上述过程，直到所得到的层数达到预期的深度。

对于 GCN 网络，给定一个核大小为 $K\times K$ 的卷积算子，以及一个通道数为 c 的输入特征映射 f_{in}，在空间位置 x 上单个通道的输出值可以写为：

$$f_{out}(x)=\sum_{h=1}^{K}\sum_{w=1}^{K} f_{in}(\mathrm{p}(x,h,\omega))\cdot \mathrm{w}(h,\omega) \tag{4-1}$$

其中，采样函数 $\mathrm{p}(x,h,\omega)$是相对于中心位置 x 在相邻像素上定义的，在图上，我们可以类似地在节点 v_{ti} 的邻居集 $B(v_{ti})=\{v_{tj}\mid d(v_{tj},v_{ti})\le D\}$ 上定义采样函数。这里 $d(v_{tj},v_{ti})$表示从 v_{tj} 到 v_{ti} 的任何路径的最小长度，因此采样函数可以写成

$$\mathrm{p}(v_{ti},v_{tj})=v_{tj} \tag{4-2}$$

类比 2D 卷积，对图中采样函数得到的邻居像素划分成不同的子集，每一个子集有一个数字标签，因此有 $l_{ti}:B(v_{ti})\to 0,...,K-1$将一个邻居节点映射到对应的子集标签，权重公式为:

$$w(v_{tj},v_{ti})=w'(l_{ti}(v_{tj})) \tag{4-3}$$

4.2.3 ST-GCN 的实现

有了精炼的采样函数和权重函数，我们现在用图卷积的形式将 GCN 网络在空间位置 x 上单个通道的输出值改写为：

$$f_{out}(v_{ti})=\sum_{v_{tj}\in B(v_{ti})}\frac{1}{Z_{ti}(v_{tj})}f_{in}(\mathrm{p}(v_{ti},v_{tj}))\cdot \mathrm{w}(v_{ti},v_{tj}) \tag{4-4}$$

其中归一化项 $Z_{ti}(v_{tj})=|\{v_{tk}\mid l_{ti}(v_{tk})=l_{ti}(v_{tj})\}|$ 等于对应子集的基数。加入这一项是为了平衡不同的贡献子集对输出的贡献。于是又得到：

$$f_{out}(v_{ti})=\sum_{v_{tj}\in B(v_{ti})}\frac{1}{Z_{ti}(v_{tj})}f_{in}(v_{tj})\cdot \mathrm{w}(l_{ti}(v_{tj})) \tag{4-5}$$

值得注意的是，如果我们将图像视为规则的 2D 网格，则该公式可以类似于标准的 2D 卷积。

随后，考虑到时空图卷积的高层次表述，我们需要对关键点进行子集划分，有以下三种方法：

（1）唯一划分

在这种策略中，每个相邻节点上的特征向量都会有一个具有相同权向量的内积。实际上，它有一个明显的缺点，在单帧情况下，使用这种策略相当于计算权向量与所有相邻节点的平均特征向量之间的内积。这对于骨架序列分类来说是次优的，因为在这个操作中可能会丢失局部微分性质。

（2）距离划分

另一种自然分区策略是根据节点的 $d(\cdot,v_{ti})$到根节点 v_{ti} 的距离，在这过程，设置 D=1，所以相邻子集将被分成两个子集，其中 d=0 指的是根节点本身，剩余的相邻节点在 d=1 子集中。因此，将有两个不同的权重向量，它们能够建模局部微分属性，如关节之间的相对平移。有 K=2 和

$$l_{ti}(v_{tj})=d(v_{tj},v_{ti}) \tag{4-6}$$

（3）空间构型划分

由于身体骨架在空间上是局部化的，仍然可以在分区过程中利用这种特定的空间配置[53]。将相邻子集划分为三个子集：第一子集连接了空间位置上比根节点更远离整个骨架的相邻节点，第二子集连接了更靠近中心的相邻

节点，第三子集为根节点本身，分别表示了离心运动、向心运动和静止的运动特征。

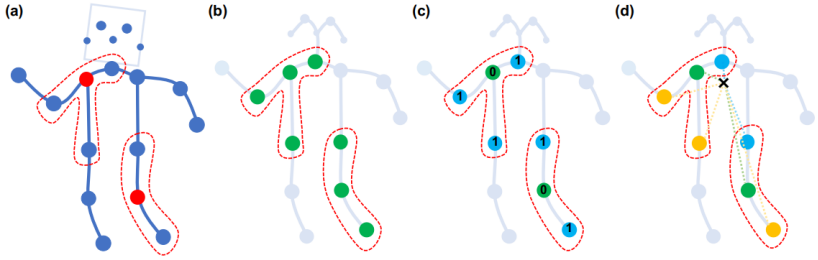

图 4.5 关键点实例(a) 划分示例(b)(c)(d)

5 基于 ST-GCN 与关键点检测的溺水检测

本文使用百度飞桨 AI Studio 平台进行模型训练和溺水检测，其配置如下表所示：

表 5.1 环境配置详情

环境配置详情	
GPU	Tesla V100
Video Mem	32GB
CPU	4Cores
RAM	32GB
Disk	100GB

本文章使用 PaddlePaddle 中的 PaddleVideo 架构和 PaddleDetection 架构。Paddle Video 是一个端到端的视频理解框架，基于 PaddlePaddle 深度学习平台实现。该框架可以支持视频分类、视频检索、视频行为识别、视频领域话题发现等多种应用场景。PaddleVideo 中包含了各种视觉和语言模型，如 ResNet、I3D、BERT 等，同时也支持多种数据增强技术和模型融合策略。这些模型和技术的组合，可以在多个数据集上实现领先水平的结果。PaddleDetection 是一个基于 PaddlePaddle 深度学习平台实现的目标检测库，它提供了丰富的目标检测算法和预训练模型，包括 Faster R-CNN、Mask R-CNN、YOLOv3、PP-YOLO 等。同时，它还为用户提供了一系列高效的目标检测工具，如数据处理工具、模型训练工具、推理部署工具等。

首先在 PaddleDetection 架构里对数据集进行关键点检测，然后得到关键点训练模型并在 PaddleVideo 里进行训练及检测，随后导出效果最好的 STGCN 模型权重，并导出 STGCN 训练好的模型导入到 PaddleDetection 架构里完成可视化检测。

5.1 数据采集与预处理

5.1.1 数据集选取

采用香港大学泳池数据集以及自行录制的正常泳姿和溺水泳姿视频。其中，香港大学泳池数据集有 120 段单目标正常泳姿视频每段视频在 30 秒左右，4 段溺水泳姿视频，每段视频在 60 秒左右。自行录制的正常泳姿和溺

水泳姿视频，分别都有两段 50 秒左右的高分辨率视频。

图 5.1 数据集部分示例

5.1.2 数据集预处理

由于实际数据中每个动作的长度不一，首先需要根据数据和实际场景预定时序长度（在 PP-Human 中我们采用 50 帧为一个动作序列），并对数据做以下处理：

（1）实际长度超过预定长度的数据，随机截取一个 50 帧的片段。

（2）实际长度不足预定长度的数据：补 0，直到满足 50 帧。

（3）恰好等于预定长度的数据：无需处理。

在本次使用的数据集里，将所有视频都截取到 50 帧以内，得到 1470 个训练视频组成训练集，300 个视频组成测试集。

5.1.3 获取序列关键点坐标

（1）数据集目标检测与关键点预推理

对于一个待标注的序列（这里序列指一个动作片段，可以是视频或有顺序的图片集合，在本文章中指训练集测试集里所有五十帧以内的视频）。可以通过模型预测或人工标注的方式获取关键点坐标，本文直接选用预推理的目标检测模型与关键点检测模型对数据集进行预处理得到序列关键点坐标。

直接选用 PaddleDetection KeyPoint 模型序列模型库中的 mot_ppyoloe_l_36e_pipeline 作为目标检测预推理模型；选择 dark_hrnet_w32_256x192 作为人体关键点预推理模型。下载模型后直接解压在模型路径，并在 PaddleDetection 中进行目标检测与关键点预推理。

每对一个视频文件进行上述目标检测与关键点预推理之后，都会得到一

个 josn 检测结果文件，将训练集与测试集的所有视频文件都进行上述目标检测与关键点预推理，将得到的 josn 文件全部都保存在一个 annotations 文件夹里。

（2）得到 PaddleVideo 可用训练文件

STGCN 是一个基于骨骼点坐标序列进行预测的模型。在 PaddleVideo 中，训练数据为采用.npy 格式存储的 Numpy 数据，标签则可以是.npy 或.pkl 格式存储的文件。对于序列数据的维度要求为(N,C,T,V,M)。维度要求以及详细说明如下表：

表 5.2 维度要求及详细说明

维度	大小	说明
N	不定	数据集序列个数
C	2	关键点坐标维度，即(x, y)
T	50	动作序列的时序维度（即持续帧数）
V	17	每个人物关键点的个数
M	1	人物个数，这里我们每个动作序列只针对单人预测

对于目标进行的关键点预推理得到的 17 个关键点序号与部位的对应关系如下表所示：

表 5.3 关键点序号对应部位

关键点序号	检测部位对应文件名称
0	nose
1	left_eye
2	right_eye
3	left_ear
4	right_ear
5	left_shoulder
6	right_shoulder
7	left_elbow
8	left_elbow
9	left_wrist

10	right_wrist
11	left_hip
12	right_hip
13	left_knee
14	right_knee
15	left_ankle
16	right_ankle

得到的各个视频的人体关键点文件形式如下图所示。

```
annotations/
├── det_keypoint_unite_image_results_fall-01-cam0-rgb.json
├── det_keypoint_unite_image_results_fall-02-cam0-rgb.json
├── det_keypoint_unite_image_results_fall-03-cam0-rgb.json
├── det_keypoint_unite_image_results_fall-04-cam0-rgb.json
    ...
├── det_keypoint_unite_image_results_fall-28-cam0-rgb.json
├── det_keypoint_unite_image_results_fall-29-cam0-rgb.json
└── det_keypoint_unite_image_results_fall-30-cam0-rgb.json
```

图 5.2 人体关键点文件形式

随后运行 prepare_dataset 脚本文件，解析每个 josn 文件内容、整理训练数据并保存数据文件。由此得到了训练数据文件 train_data 和标签文件 train_label。需要注意的是，我们在本论文中使用 PaddleDetection 进行最终的溺水检测时，规定溺水行为的标签为“1”，正常游泳的行为标签为“0”。而我们默认得到的标签文件里面，所有的标签默认为 0，故需要修改所有溺水行为的训练标签文件为 1。

5.2 人体关键点检测与姿态估计

选用 PaddleDetection KeyPoint 模型序列模型库中的 mot_ppyoloe_l_36e_pipeline 作为目标检测预推理模型；选择 dark_hrnet_w32_256x192 作为人体关键点预推理模型。导入训练数据进行目标检测与关键点预推理，在 PaddleDetection 组件下的输出文件效果如下图所示。

图 5.3 关键点检测实例

所有视频中人体的十七个关键点检测结果良好。

5.3 模型训练

在 PaddleVideo 路径下运行 main.py 文件运用的人体关键点预推理检测模型 stgcn_pphuman.yaml。在训练的同时开启验证，保存效果最佳的模型文件以及模型权重，方便之后的模型检验以及检测可视化。

模型训练时，会返回每一次训练结束后的 loss（损失值）、ir（当前学习率）、batch cost（数据处理时间）、ips（每秒训练样本数）等数值。其中，得到模型训练后总体平均损失值为 0.1865，损失值在合理范围内，模型效果良好且不会出现过拟合的情况。

```
[05/01 17:17:30] epoch:[ 33/500 ] train step:0    loss: 0.11768 lr: 0.014356 top1: 1.00000 top5: 0.00000 batch_cost: 0.38767 sec, reader_cost: 0.34970 sec, ips: 5.15905 instance/sec, eta: 0:00:00
[05/01 17:17:30] END epoch:33  train loss_avg: 0.16877  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02648 sec, avg_reader_cost: 0.00027 sec, batch_cost_sum: 0.44342 sec, avg_ips: 13.5
[05/01 17:17:31] epoch:[ 34/500 ] train step:0    loss: 0.13281 lr: 0.012956 top1: 1.00000 top5: 0.00000 batch_cost: 0.38858 sec, reader_cost: 0.35178 sec, ips: 5.14693 instance/sec, eta: 0:00:00
[05/01 17:17:31] END epoch:34  train loss_avg: 0.24613  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02517 sec, avg_reader_cost: 0.00031 sec, batch_cost_sum: 0.43996 sec, avg_ips: 13.6
[05/01 17:17:31] epoch:[ 35/500 ] train step:0    loss: 0.13035 lr: 0.011604 top1: 1.00000 top5: 0.00000 batch_cost: 0.36632 sec, reader_cost: 0.33014 sec, ips: 5.45966 instance/sec, eta: 0:00:00
[05/01 17:17:31] END epoch:35  train loss_avg: 0.27130  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02472 sec, avg_reader_cost: 0.00026 sec, batch_cost_sum: 0.41894 sec, avg_ips: 14.3
[05/01 17:17:32] epoch:[ 36/500 ] train step:0    loss: 0.30591 lr: 0.010305 top1: 1.00000 top5: 0.00000 batch_cost: 0.39124 sec, reader_cost: 0.35507 sec, ips: 5.11196 instance/sec, eta: 0:00:00
[05/01 17:17:32] END epoch:36  train loss_avg: 0.27997  top1_avg: 0.83333 top5_avg: 0.00000 avg_batch_cost: 0.02319 sec, avg_reader_cost: 0.00022 sec, batch_cost_sum: 0.43904 sec, avg_ips: 13.6
[05/01 17:17:32] epoch:[ 37/500 ] train step:0    loss: 0.09640 lr: 0.009064 top1: 1.00000 top5: 0.00000 batch_cost: 0.40646 sec, reader_cost: 0.36902 sec, ips: 4.92058 instance/sec, eta: 0:00:00
[05/01 17:17:32] END epoch:37  train loss_avg: 0.10292  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02467 sec, avg_reader_cost: 0.00028 sec, batch_cost_sum: 0.45720 sec, avg_ips: 13.1
[05/01 17:17:33] epoch:[ 38/500 ] train step:0    loss: 0.56804 lr: 0.007886 top1: 1.00000 top5: 0.00000 batch_cost: 0.41743 sec, reader_cost: 0.37959 sec, ips: 4.79118 instance/sec, eta: 0:00:00
[05/01 17:17:33] END epoch:38  train loss_avg: 0.28270  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02699 sec, avg_reader_cost: 0.00031 sec, batch_cost_sum: 0.47184 sec, avg_ips: 12.7
[05/01 17:17:34] epoch:[ 39/500 ] train step:0    loss: 0.14883 lr: 0.006776 top1: 1.00000 top5: 0.00000 batch_cost: 0.42157 sec, reader_cost: 0.38307 sec, ips: 4.74413 instance/sec, eta: 0:00:00
[05/01 17:17:34] END epoch:39  train loss_avg: 0.11365  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02794 sec, avg_reader_cost: 0.00029 sec, batch_cost_sum: 0.47821 sec, avg_ips: 12.5
[05/01 17:17:34] epoch:[ 40/500 ] train step:0    loss: 0.15064 lr: 0.005737 top1: 1.00000 top5: 0.00000 batch_cost: 0.39193 sec, reader_cost: 0.35343 sec, ips: 5.10296 instance/sec, eta: 0:00:00
[05/01 17:17:34] END epoch:40  train loss_avg: 0.11010  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02796 sec, avg_reader_cost: 0.00035 sec, batch_cost_sum: 0.44737 sec, avg_ips: 13.4
[05/01 17:17:35] epoch:[ 41/500 ] train step:0    loss: 0.08702 lr: 0.004775 top1: 1.00000 top5: 0.00000 batch_cost: 0.41139 sec, reader_cost: 0.37331 sec, ips: 4.86161 instance/sec, eta: 0:00:00
[05/01 17:17:35] END epoch:41  train loss_avg: 0.09729  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02552 sec, avg_reader_cost: 0.00046 sec, batch_cost_sum: 0.46606 sec, avg_ips: 12.8
[05/01 17:17:35] epoch:[ 42/500 ] train step:0    loss: 0.08579 lr: 0.003892 top1: 1.00000 top5: 0.00000 batch_cost: 0.39200 sec, reader_cost: 0.35246 sec, ips: 5.10198 instance/sec, eta: 0:00:00
[05/01 17:17:36] END epoch:42  train loss_avg: 0.08278  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02335 sec, avg_reader_cost: 0.00025 sec, batch_cost_sum: 0.44249 sec, avg_ips: 13.5
[05/01 17:17:36] epoch:[ 43/500 ] train step:0    loss: 1.89546 lr: 0.003092 top1: 0.50000 top5: 0.00000 batch_cost: 0.42407 sec, reader_cost: 0.38274 sec, ips: 4.71617 instance/sec, eta: 0:00:00
[05/01 17:17:36] END epoch:43  train loss_avg: 0.68879  top1_avg: 0.83333 top5_avg: 0.00000 avg_batch_cost: 0.02532 sec, avg_reader_cost: 0.00030 sec, batch_cost_sum: 0.47642 sec, avg_ips: 12.5
[05/01 17:17:37] epoch:[ 44/500 ] train step:0    loss: 0.07894 lr: 0.002379 top1: 1.00000 top5: 0.00000 batch_cost: 0.39183 sec, reader_cost: 0.35455 sec, ips: 5.10423 instance/sec, eta: 0:00:00
[05/01 17:17:37] END epoch:44  train loss_avg: 0.08335  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02752 sec, avg_reader_cost: 0.00028 sec, batch_cost_sum: 0.44548 sec, avg_ips: 13.4
[05/01 17:17:37] epoch:[ 45/500 ] train step:0    loss: 0.14480 lr: 0.001756 top1: 1.00000 top5: 0.00000 batch_cost: 0.39921 sec, reader_cost: 0.36161 sec, ips: 5.00984 instance/sec, eta: 0:00:00
[05/01 17:17:38] END epoch:45  train loss_avg: 0.10737  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02736 sec, avg_reader_cost: 0.00032 sec, batch_cost_sum: 0.45396 sec, avg_ips: 13.2
[05/01 17:17:38] epoch:[ 46/500 ] train step:0    loss: 0.36612 lr: 0.001224 top1: 1.00000 top5: 0.00000 batch_cost: 0.38667 sec, reader_cost: 0.34987 sec, ips: 5.17239 instance/sec, eta: 0:00:00
[05/01 17:17:38] END epoch:46  train loss_avg: 0.23349  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02789 sec, avg_reader_cost: 0.00035 sec, batch_cost_sum: 0.44110 sec, avg_ips: 13.6
[05/01 17:17:39] epoch:[ 47/500 ] train step:0    loss: 0.07896 lr: 0.000785 top1: 1.00000 top5: 0.00000 batch_cost: 0.38876 sec, reader_cost: 0.35018 sec, ips: 5.14452 instance/sec, eta: 0:00:00
[05/01 17:17:39] END epoch:47  train loss_avg: 0.09852  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02836 sec, avg_reader_cost: 0.00030 sec, batch_cost_sum: 0.44324 sec, avg_ips: 13.5
[05/01 17:17:39] epoch:[ 48/500 ] train step:0    loss: 0.36706 lr: 0.000443 top1: 1.00000 top5: 0.00000 batch_cost: 0.38629 sec, reader_cost: 0.34595 sec, ips: 5.17751 instance/sec, eta: 0:00:00
[05/01 17:17:39] END epoch:48  train loss_avg: 0.20610  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02756 sec, avg_reader_cost: 0.00033 sec, batch_cost_sum: 0.44244 sec, avg_ips: 13.5
[05/01 17:17:40] epoch:[ 49/500 ] train step:0    loss: 0.15069 lr: 0.000197 top1: 1.00000 top5: 0.00000 batch_cost: 0.38819 sec, reader_cost: 0.34840 sec, ips: 5.15208 instance/sec, eta: 0:00:00
[05/01 17:17:40] END epoch:49  train loss_avg: 0.10248  top1_avg: 1.00000 top5_avg: 0.00000 avg_batch_cost: 0.02909 sec, avg_reader_cost: 0.00028 sec, batch_cost_sum: 0.44684 sec, avg_ips: 13.4
```

图 5.4 部分训练结果

模型训练结束后，得到 output 路径下的模型文件以及相关的模型权重以供后续导出模型及权重。在此，训练 1470 个训练视频组成训练集，其中有 500 个为溺水姿态构成的视频文件，标注文件为 1；有 970 个正常泳姿构成的视频文件，标注文件为 0。

5.4 模型测试

在数据预处理时，已将测试集进行同样的预处理操作，并且将溺水的测试集视频标注文件改为 1，正常泳姿的标注文件默认不变为 0。

模型训练完成后，在 PaddleVideo 路径下进行模型检测，ST-GCN 模型权重选用训练效果最好的模型权重 STGCN_best.pdparams。

在进行最终的检测时，通过得到模型的行为权重与输入视频的人体关键点的拟合程度，对比得到最高的拟合程度，从而识别为相应的行为。反馈部分拟合结果如下图所示。

```
Tensor(shape=[1, 2], dtype=float32, place=Place(gpu:0), stop_gradient=True,
       [[ 0.17297399, -0.24139857]])
Tensor(shape=[1, 2], dtype=float32, place=Place(gpu:0), stop_gradient=True,
       [[ 0.57651079, -0.60299206]])
Tensor(shape=[1, 2], dtype=float32, place=Place(gpu:0), stop_gradient=True,
       [[-1.30350351,  1.08959544]])
Tensor(shape=[1, 2], dtype=float32, place=Place(gpu:0), stop_gradient=True,
       [[ 0.64776915, -0.66688430]])
Tensor(shape=[1, 2], dtype=float32, place=Place(gpu:0), stop_gradient=True,
       [[-0.82808942,  0.65948308]])
```

图 5.5 输入视频与模型训练行为拟合结果

最终，模型的测试结果为：返回的 top1 值为 0.95，该单目标检测模型具有良好的检测效果。

5.5 模型检测结果可视化

上述过程我们得到了检测效果良好的溺水检测模型以及相关的模型权重。但是由于在使用过程中，不可能将每一段视频都截取为一段一段五十帧的视频然后再导入检测，这样检测效率太低。于是我们便要将训练好的模型以及相关的模型权重导出。导入到 PaddleDetection 架构中，使用 PaddleDetection 对一段视频进行持续的溺水检测，以代替实时监测的效果。

5.5.1 模型导出

在 PaddleVideo 中，通过实现模型的导出，得到模型结构文件 STGCN.pdmodel 和模型权重文件 STGCN.pdiparams，并增加配置文件。同时要根据 PaddleDetection 路径下的 PP-human 调用 ST-GCN 模型及权重的格式，修改经过 PaddleVideo 训练过的 ST-GCN 模型及权重文件名。最后移动到 PaddleDetection 的路径下，替换相关的检测模型及权重。完成后的导出模型结构如下图。

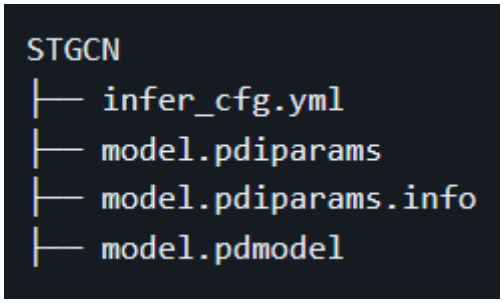

图 5.6 导出模型结构

5.5.2 自定义修改输出并修改可视化输出

基于人体骨骼点的行为识别方案中，模型输出的分类结果即代表了该人物在一定时间段内行为类型。对应分类的类型最终即视为当前阶段的行为。因此在完成自定义模型的训练及部署的基础上，使用模型输出作为最终结果，修改可视化的显示结果即可。在 PaddleDetection 路径下，将输出文件中提示框输出改为“Drowning”，代表检测出了溺水行为。

5.5.3 利用 PaddleDetection 进行溺水检测结果可视化

根据 PaddleDetection 路径下的 PP-human 调用 ST-GCN 模型及权重的格式，修改经过 PaddleVideo 训练过的 ST-GCN 模型及权重文件名。最后移动到 PaddleDetection 的路径下，替换相关的检测模型及权重。由此便可以在 PaddleDetection 架构下对视频进行持续的溺水检测。

采用香港大学数据集中一段既有正常泳姿又有溺水泳姿的未经训练与检测的五十秒视频导入到 PaddleDetection 进行检测，反馈出检测视频到 output 文件夹里。节选正常泳姿的检测结果如下图，检测并跟踪到人体，检测出人体关键点，识别为正常游泳，不弹出任何提示框。

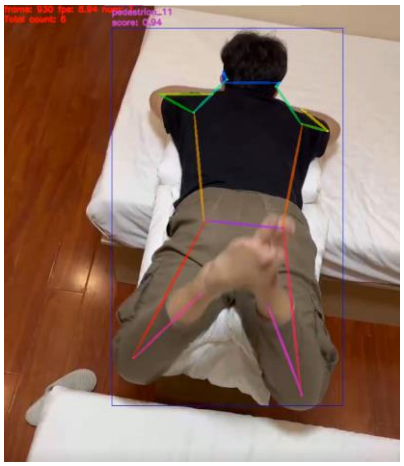

图 5.7 节选正常泳姿检测结果

节选检测到溺水的结果如下图所示，在检测到游泳者出现溺水行为时，会出现“Drowning”提示框。可见，PaddleDetection 能较为准确地检测出单目标的正常泳姿与溺水泳姿。

图 5.8 节选溺水泳姿检测结果

6 总结与展望

本文阐述了目标检测、关键点检测与行为识别以及溺水检测的研究现状与应用，介绍了基于 ST-GCN 与关键点检测的行为识别方法。通过录制实际的游泳视频，并搜集到香港大学泳池数据集，对采集到的视频数据进行预处理。使用百度飞桨 AI Studio 平台的 PaddleVideo 架构对数据集进行目标检测及关键点预推理，得到训练效果良好的 ST-GCN 模型及模型权重，将此训练好的模型及模型权重导入到 PaddleDetection 架构中，实现了对视频内单目标的溺水识别。

通过模型训练与测试，PaddleVideo 架构中的 ST-GCN 模型对室内游泳馆游泳者的溺水检测实现了良好的效果，根据测试集的检测，其检测准确度达到了 0.95，检测效果良好，误报和漏报率维持在合理范围内。在使用 PaddlePaddle 平台里的 PaddleDetection 架构进行对视频的持续溺水检测时，对单目标的跟踪效果优秀，系统的检测速率良好，可以满足室内游泳馆游泳者的跟踪及溺水检测。

虽然本文在对室内游泳馆的单目标跟踪与溺水检测中取得了不错的效果，但也存在一些不足之处：首先，本文采用经过 PaddleVideo 架构训练效果良好的 ST-GCN 模型及模型权重直接导入到 PaddlePaddle 平台里的 PaddleDetection 架构进行对视频的持续溺水检测，导致输出的检测视频里面检测到溺水行为时总是会慢五十帧弹出提示框，恢复正常泳姿时，溺水提示框也会延迟五十帧消失。由于游泳时每个人的游泳状态是不一样的，所以会出现误报漏报的情况。同时，现阶段本论文仅做出针对室内游泳馆单目标的溺水检测，没有实现多目标的溺水检测。还存在很大的提升空间。

参考文献

[1] 王仰江, 刘伟, 姚兴勇. 关于防溺水汽车浮力装置的研究[J]. 自动化与仪器仪表, 2019(4):30-33.

[2] GUNNAR, JOHANSSON. Visual Perception of Biological Motion and A Model for Its Analysis[J]. Attention, Perception & Psychophysics, 1973,14(2):201-211.

[3] 刘晓悦, 王云明. 基于 HOG-SVM 的改进跟踪-学习-检测算法的目标跟踪方法[J].科学技术与工程,2019,19(27):266-271.

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

[5] Tan Chenjiao, Li Changying, He Dongjian, Song Huaibo. Towards real-time tracking and counting of seedlings with a one-stage detector and optical flow[J]. Computers and Electronics in Agriculture.2022,193(2):1-12.

[6] Liang Tianjiao, Bao Hong, Pan Weiguo, Fan Xinyue, Li Han. AspectNet: Aspect-Aware Anchor-Free Detector for Autonomous Driving[J]. Applied Sciences,2022,12(12)：5972-5972.

[7] Oliver Matthieu, Renou Amélie, Allou Nicolas, Moscatelli Lucas, Ferdynus Cyril, Allyn Jerôme. Image augmentation and automated measurement of endotracheal-tube-to-carina distance on chest radiographs in intensive care unit using a deep learning model with external validation.[J]. Critical care (London, England),2023,27(1):17-20.

[8] Saleh Mubarak Auwalu, Ameen Zubaida Said, Altrjman Chadi, AlTurjman Fadi. Computer-Vision-Based Statue Detection with Gaussian Smoothing Filter and EfficientDet[J]. Sustainability,2022,14(18):11413-11413.

[9] 史凌凯, 耿毅德, 王宏伟, 王洪利. 基于改进 Mask R-CNN 的刮板输送机铁质异物多目标检测[J].工矿自动化,2022,48(10):55-61.

[10] 伍锡如, 邱涛涛, 王耀南. 改进 Mask R-CNN 的交通场景多目标快速检测与分割[J].仪器仪表学报,2021,42(07):242-249.

[11] 何止戈. 基于深度学习方法的 PCB 图像缺陷检测[D].电子科技大学,2020.

[12] 徐守坤, 顾佳楠, 庄丽华, 李宁, 石林, 刘毅. 基于两阶段计算 Transformer 的小目标检测[J].计算机科学与探索:2023,12(2):1-21.

[13] Maktab Dar Oghaz Mahdi,Razaak Manzoor, Remagnino Paolo. Enhanced Single Shot Small Object Detector for Aerial Imagery Using Super-Resolution, Feature Fusion and Deconvolution.[J]. Sensors (Basel, Switzerland),2022,22(12):4339-4339.

[14] Young-Joon Hwang, Jin-Gu Lee, Un-Chul Moon, Ho-Hyun Park. SSD-TSEFFM: New SSD Using Trident Feature and Squeeze and Extraction Feature Fusion[J]. Sensors,2020,20(13):3630-3630.

[15] 纪超群. 基于深度学习的道路场景轻量级目标检测算法研究[D].长春工业大学,2022.

[16] 张诗慧. 基于 RetinaNet 的高铁无砟轨道板表面裂缝检测方法研究[D].华东交通大学,2022.

[17] 仲鹏宇, 杨娟. 基于轻量级双模态 SSD 算法的疲劳驾驶检测[J].电子技术与软件工程,2022(03):145-149.

[18] Feng Xiaoxu, Yao Xiwen, Shen Hui, Cheng Gong, Xiao Bin, Han Junwei. Learning an Invariant and Equivariant Network for Weakly Supervised Object Detection.[J]. IEEE transactions on pattern analysis and machine intelligence,2023,15(12):13-15.

[19] Sangineto Enver, Nabi Moin, Culibrk Dubravko, Sebe Nicu. Self Paced Deep Learning for Weakly Supervised Object Detection.[J]. IEEE transactions on pattern analysis and machine intelligence,2019,41(3):15-17.

[20] 曾文献, 马月, 李伟光. 轻量化二维人体骨骼关键点检测算法综述[J].科学技术与工程,2022,22(16):6377-6392.

[21] 刘圣杰, 何宁, 于海港, 王程, 韩文静. 引入坐标注意力和自注意力的人体关键点检测研究[J].计算机工程,2022,48(12):86-94.

[22] Huang Ying, Huang He. Stacked attention hourglass network based robust facial landmark detection.[J]. Neural networks : the official journal of the International Neural Network Society,2022,157(2):65-67.

[23] 项超. 无人机系统下视频图像人体姿态分析的研究[D].南京航空航天大学,2020.

[24] Ji Xiaodong, Yang Qiaoning, Yang Xiuhui, Zheng Jiahao, Gong Mengyan. Human Pose Estimation: Multi-stage Network Based on HRNet[J]. Journal of Physics: Conference Series,2022,2400(1):115-118.

[25] Li Lei, Hassan Muhammad Adeel, Yang Shurong, Jing Furong, Yang Mengjiao, Rasheed Awais, Wang Jiankang, Xia Xianchun, He Zhonghu, Xiao Yonggui. Development of image-based wheat spike counter through a Faster R-CNN algorithm and application for genetic studies[J]. The Crop

Journal,2022,10(5):55-59.

[26] Sahoo Pravat Kumar, Mishra Sushruta, Panigrahi Ranjit, Bhoi Akash Kumar, Barsocchi Paolo. An Improvised Deep-Learning-Based Mask R-CNN Model for Laryngeal Cancer Detection Using CT Images[J]. Sensors,2022,22(22):46-49.

[27] Sahin Ipsita, Modi Arjun, Kokkoni Elena. Evaluation of OpenPose for Quantifying Infant Reaching Motion[J]. Archives of Physical Medicine and Rehabilitation,2021,102(10):15-18.

[28] Mohammed Boutalline, Adil Tannouche, Hassan Faouzi, Hamid Ouanan, Malak Dargham. Automatic Detection and Classification of Apple Leaves Diseases Using MobileNet V2[J]. Revue d'Intelligence Artificielle,2022,36(5):45-49.

[29] 邢仁琦, 杨怀志, 薄一军, 尤嘉, 张淳杰, 李丹勇. 基于轻量化 EfficientNet 的小目标裂缝检测算法[J/OL].北京交通大学学报:1-8[2023-05-19].

[30] Angeles Ceron Juan Carlos, Chang Leonardo, Ruiz Gilberto Ochoa, Ali Sharib. Assessing YOLACT++ for real time and robust instance segmentation of medical instruments in endoscopic procedures.[J]. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference,2021,28(4):25-29.

[31] 王素, 王嘉澳, 单大国. 基于不变矩改进 HMM 的人体异常行为识别研究[J].广东公安科技,2022,30(04):45-49.

[32] 武东辉, 许静, 陈继斌, 孙彦玺, 仇森. 基于融合注意力机制与 CNN-LSTM 的人体行为识别算法[J].科学技术与工程,2023,23(02):681-689.

[33] Anh-Vu Bui, Thi-Oanh Nguyen. Multi-view Human Action Recognition Based on TSN Architecture Integrated with GRU[J]. Procedia Computer Science,2020,176(02):48-51.

[34] Mazzia Vittorio, Angarano Simone, Salvetti Francesco, Angelini Federico, Chiaberge Marcello. Action Transformer: A self-attention model for short-time pose-based human action recognition[J]. Pattern Recognition,2022,124(02):15-19.

[35] Alfasly Saghir, Chui Charles K,Jiang Qingtang, Lu Jian, Xu Chen. An Effective Video Transformer With Synchronized Spatiotemporal and Spatial Self-Attention for Action Recognition.[J]. IEEE transactions on neural networks and learning systems,2022,189(04):18-25.

[36] Zin Thi Thi, Htet Ye, Akagi Yuya, Tamura Hiroki, Kondo Kazuhiro, Araki Sanae, Chosa Etsuo. Real-Time Action Recognition System for Elderly

People Using Stereo Depth Camera[J]. Sensors,2021,21(17):18-23.

[37] Umar Iqbal, Anton Milan, Juergen Gall. Pose-Track: Joint Multi-Person Pose Estimation and Tracking.[J]. CoRR,2016,161(2):15-19.

[38] W. Dahl. Perceptions of Evidence-based Dietetic Practice, Attitudes, and Abilities of MS-DI Program Graduates[J]. Journal of the Academy of Nutrition and Dietetics,2019,119(9):15-18.

[39] YAN S, SMITH J S, LU W, et al. CHAM: Action Recognition Using Convolutional Hierarchical Attention Model[C] // 2017 IEEE International Conference on Image Processing (ICIP). 2017 :3958 – 3962.

[40] SHARMA S, KIROS R, SALAKHUTDINOV R. Action Recognition using Visual Attention[J].ArXiv, 2015, abs/1511.04119.

[41] YAN S, XIONG Y, LIN D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition[C] // AAAI. 2018:7444–7452.

[42] 瑞士 BlueFox 泳池安全系统工作原理简介[J].游泳,2015(02):31.

[43] Peel Lauren R, Daly Ryan, Keating Daly Clare A, Stevens Guy M W, Collin Shaun P, Meekan Mark G. Stable isotope analyses reveal unique trophic role of reef manta rays (Mobula alfredi) at a remote coral reef.[J]. Royal Society open science,2019,6(9):1-9.

[44] 吴婷璇. 基于视频监控的游泳者检测与跟踪. 太原理工大学. 2016.

[45] 乔羽. 基于 Mask R-CNN 泳池中溺水行为检测系统的设计与实现. 青岛大学. 2019.

[46] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation[C/OL]//Leibe B,Matas J,Sebe N,et al.Computer Vision–ECCV 2016.Cham:Springer Interna-tional Publishing,2016,(2):483-499.

[47] Chen Y, Wang Z, Peng Y, et al. Cascaded pyramid network for multi-person pose es-timation[C/OL]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recogni-tion(CVPR).Los Alamitos,CA,USA:IEEE Computer Society,2018:7103-7112.

[48] Xiao B, Wu H, Wei Y. Simple baselines for human pose estimation and tracking[C/OL]//Ferrari V,Hebert M,Sminchisescu C,et al.Computer Vision–ECCV 2018.Cham:SpringerInternational Publishing,2018:472-487.

[49] Insafutdinov E, Pishchulin L, Andres B, et al. Deepercut:A deeper,stronger,and faster multi-person pose estimation model[C]//European conference on computer vision.Springer,2016:34-50.

[50] 陈曦. 基于多尺度方法及小样本学习的计算机辅助诊断方法与应用[D]. 桂林电子科技大学,2021.

[51] 庞江淼. 基于深度学习的光学图像场景感知算法研究[D].浙江大学,2021.

[52] 张赟疆. 面向医学应用的人体关键点检测技术研究[D].电子科技大学,2022.

[53] 于国旺.一种基于深度学习的平斜腕臂连接处螺栓缺失检测方法[J].电气化铁道,2022,33(S1):76-80.

[54] 郭澄霖. 基于深度学习的火焰检测与边缘计算设备部署[D].海南大学,2022.

在学取得成果

一、 在学期间所获的奖励

2020.04 获新生三等奖学金 北京科技大学教务处

2021.04 获人民一等奖学金 北京科技大学教务处

2021.10 获 ICAN 创新创业北京赛区二等奖 北京市教育局

2021.11 获北京科技大学智能车竞赛一等奖 北京科技大学

2022.04 获人民三等奖学金 北京科技大学教务处

2022.09 获北京科技大学数学竞赛二等奖 北京科技大学教务处

二、 在学期间发表的论文

李荣斌,陈毛,张少军,许记雷,袁浩天,杨沛胥,刘风琴.硅热法炼镁还原过程熔融黏结机理及控制[J].有色金属(冶炼部分),2022(06):30-36.

三、 在学期间取得的科技成果

无

致 谢

行文至此落笔中，始于初春终于夏。

四年时间很短，行将毕业，大学的一切都历历在目。在北京科技大学的四年时间，遇到的每一位老师、同学，上过的每一节课，拼搏过的每一场比赛，参加过的每一次活动，见证了我这四年来的点点滴滴，感谢学校，感谢高等工程师学院（现卓越工程师学院），感谢所有的老师、辅导员、教务人员，你们对我的栽培，都会是我不可多得且永远珍视的人生经历。

“人生之幸，得遇良师。”感谢我的导师——张老师、艾老师，老师学识渊博，为学严谨认真，待人和蔼可亲。这四年里，老师们对我的学业、科研、生活悉心指导和帮助，是我大学的引路人，是我的当良师益友。

“树高千尺不忘根深洪土。”感谢我的家人，一直在用自己的方式全心全意地爱我、尊重并支持我成长路上的每一个选择，感谢他们无条件地选择相信我，支持我。养育之恩，无以为报。

“上感君犹念，傍惭友或推。”感谢我的女朋友张冰焱同学，在我疲惫的时候给予鼓励，在我迷茫的时候陪伴安慰你的日子，我平凡的每一天都充满了无尽的幸福。

感谢六斋 112 的室友们、高工 1902 的全体同学一直以来给予的关心与帮助，感谢相遇。

感谢陆俊达学长，在我的整个毕业设计工作中给予了十分重要的帮助，感谢学长点拨，让我在研究方法方面一次次取得突破。

本科阶段结束，即将进入研究生阶段，翻开崭新的一页，我满怀期待，我将以更加认真的态度、更加充沛的精力来完成研究生的学习工作。