

基于YOLO的公共区域人群社交距离检测方法

陈冰, 戴世亮, 叶彬雅

浙江工业大学之江学院, 信息工程学院, 浙江 绍兴

收稿日期: 2023年6月11日; 录用日期: 2023年7月28日; 发布日期: 2023年8月7日

摘要

针对人流密集的公共区域, 本论文提出了一种基于YOLO模型的人群社交距离检测方法, 旨在进行有效的人流控制和预防疫情传播。该方法首先利用YOLO模型对公共区域的视频进行行人检测, 获取每个行人的位置关键点信息。接着, 通过KD-tree算法检测每个行人的邻居行人, 并估计它们之间的距离。根据预设的安全距离阈值, 对行人的社交距离进行风险评估, 包括危险、安全或需要注意的状态。为了更好地呈现结果, 本文将危险状态标识为红色危险标识, 安全状态标识为绿色安全标识, 而中间状态则标识为紫色注意标识。实验结果表明, 所提出的行人社交距离检测方法具有较高的实时性和准确性。

关键词

YOLO模型, 社交距离, 目标检测, 风险评估

A Social Distancing Detection Method for Public Area Crowds Based on YOLO

Bing Chen, Shiliang Dai, Binya Ye

School of Information Engineering, Zhijiang College of Zhejiang University of Technology, Shaoxing Zhejiang

Received: Jun. 11th, 2023; accepted: Jul. 28th, 2023; published: Aug. 7th, 2023

Abstract

This paper presents a crowd social distancing detection method based on the YOLO model, aiming to effectively control crowd movement and prevent the transmission of infectious diseases in densely populated public areas. The proposed method leverages the YOLO model to perform pedestrian detection on videos captured in public areas, thereby extracting precise positional key

point information for each individual. Subsequently, the KD-tree algorithm is employed to identify neighboring pedestrians for each detected individual and estimate the inter-person distances between them. By utilizing a predefined threshold for a safe distance, a risk assessment is conducted to categorize the social distancing status of pedestrians into dangerous, safe, or attention-required states. To enhance the interpretability of the results, this study employs a red danger symbol to represent the dangerous state, a green safety symbol to represent the safe state, and a purple attention symbol to represent the intermediate state. Experimental results demonstrate that the proposed crowd social distancing detection method achieves high real-time performance and accuracy in assessing and monitoring social distancing compliance among pedestrians.

Keywords

YOLO Model, Social Distance, Object Detection, Risk Assessment

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

全球范围内新型冠状病毒肺炎(COVID-19)的爆发和传播使公共区域的人群密集场景成为疫情防控的重点关注领域。在这种背景下,为了有效控制人流并减少人与人之间的接触,保持适当的社交距离变得至关重要。然而,传统的监控和管理手段在监测人群社交距离方面存在许多限制和困难,无法满足实时、准确和全面监测的需求。尤其在公共场所,人流量巨大,通过传统的人力劝导行人保持安全间距不仅需要消耗大量的人力资源和时间,而且效果有限,无法高效地疏散人群。因此,迫切需要在关键的公共区域部署一套可靠的社交距离检测方法来应对这些挑战。

为了达到安全的社交距离,需要通过保持个体之间的物理距离和减少人群之间的接触来减缓或阻止病毒的传播。然而,在校园、商业中心、工厂等人流密集的场所,时刻保持安全距离是一项具有一定困难的任务。为了应对这一挑战,需要将人工智能和深度学习技术集成到关键区域的监控系统中,以实现智能的行人社交距离检测。王林等[1]提出一种基于轻量化神经网络的距离检测系统用于检测社交距离,该系统跟踪准确率可达到 91.04%,但由于 YOLOv5 显著改善 YOLO 对象检测的性能,因此前者并没有广泛运用于市场;王鹏鑫等[2]从数据处理、网络结构、损失函数等多方面入手提出一系列对目标检测算法的改进方式,以提高检测效率和准确度;许德刚等[3]利用提升 NMS 算法的运算效率和选取精度的方法来改进预测框选取方法不精确的问题;王婧媛等[4]提出在拥挤场所识别多个目标的检测系统,但损失函数需考虑检测框重叠部分距离信息判定的问题,故准确度不高。

为了解决人流密集的公共区域中的社交距离控制挑战,本文旨在提出一种基于 YOLO 模型的人群社交距离检测方法。该方法融合了目标检测和距离估计的技术,通过利用 YOLO 模型对公共区域的视频进行行人检测,能够准确地定位和识别每个行人的位置信息。随后,采用 KD-tree 算法来检测行人之间的邻居关系,并通过距离度量方法来估计它们之间的距离,从而实现对行人社交距离的精细分析。

本文的组织结构如下:第 2 节详细介绍了基于 YOLO 模型的行人社交距离估计方法。该方法包括 YOLO 行人目标检测模型的训练过程,行人关键点的提取方法以及基于 KD-tree 算法的社交距离估计方法。在第 3 节中,描述了实验设置和评估指标,以验证所提出方法的性能分析。最后,在总结部分对本文的研究内容进行归纳总结,并对未来的研究方向进行展望。

2. 基于 YOLO 的行人社交距离估计

2.1. 训练 YOLO 行人识别模型

YOLOv5 [5] (You Only Look Once)是由 Ultralytics LLC 公司于 2020 年 5 月提出, 图像推理的速度最快达 0.007 s, 即每秒可以处理 140 帧, 满足视频图像实时检测需求, 同时结构小巧, YOLOv5s 版本的权重数据为 YOLOv4 的 1/9, 大小为 27 MB。其网络基本架构图如图 1 所示。其网络模型结构由四部分构成: 输入端、Backbone、Nect、Prediction。

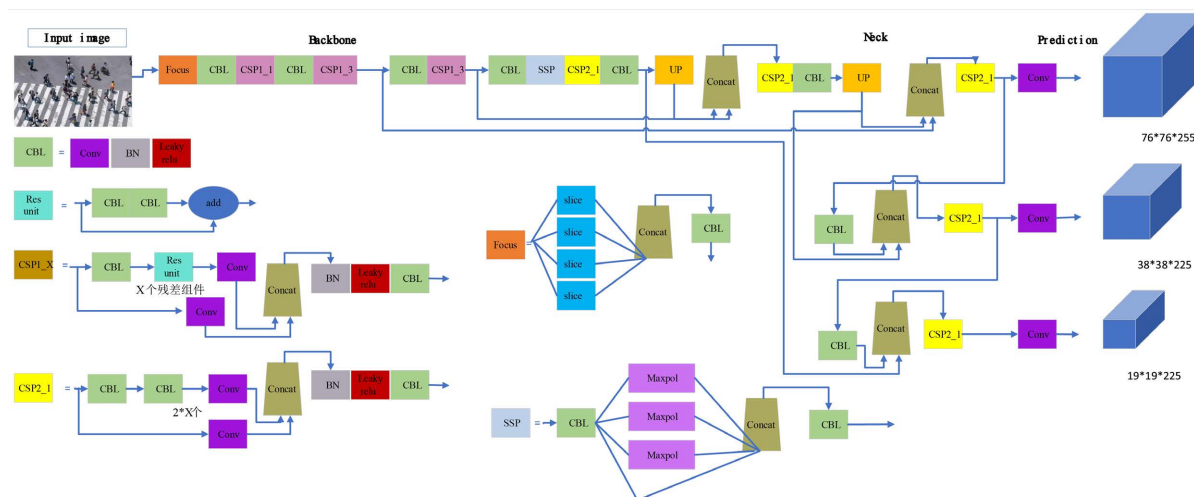


Figure 1. The basic architecture of YOLOv5

图 1. YOLOv5 基本架构

本文采用 YOLOv5 算法, 其中 Backbone 不同于其他版本的地方是多了 Focus 层, 如图 1 所示。对输入的图片进行切片操作, 优点在于可以从高分辨率的图像中提取像素点转移到低分辨率的图像中, 扩大像素点的感受野(卷积神经网络每一层输出的特征图), 可降低模型的计算量, 提高工作效率。输入端提出自适应锚框计算、图像大小缩放以及 Mosaic 数据增强。其模型的运行如图 2 所示。

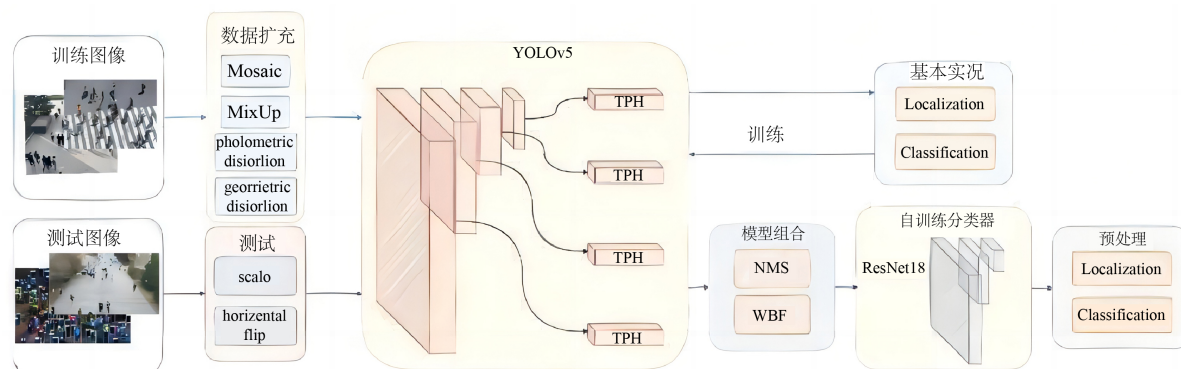


Figure 2. YOLO operation diagram

图 2. YOLO 工作流程

本文需要使用自定义数据集进行 YOLO 模型训练以便于提高行人检测的精确度。具体步骤如下:

步骤一：数据收集和标注。从网络上收集行人相关的图像数据，并使用标注工具 LabelImg 进行行人标注。标注的目的是为每个图像中的目标对象(如行人)标记边界框和类别标签。此外，本文只需识别“person”这个类别，即单类别学习。

步骤二：划分训练集和验证集。将收集的数据集划分为训练集和验证集。训练集用于模型参数的优化和训练过程，验证集用于评估模型在未见过的数据上的性能表现。在 YAML 文件中配置训练集和验证集的路径信息。

步骤三：配置模型和参数。在进行训练之前，在 YAML 文件中配置 YOLO 模型的结构和相关参数，同时设置模型训练的超参数，如学习率、批大小、迭代次数等。

步骤四：模型训练。使用划分好的训练集数据进行模型训练。基于 Pytorch 深度学习框架，使用 GPU 对标注后的图像对 YOLO 模型进行迁移学习和训练，通过反向传播和梯度下降算法来更新模型的权重。

2.2. 行人关键点提取

在行人关键点提取[6]过程中，首先根据训练好的 YOLO 模型对待识别的图像数据进行行人检测。对于每个检测到的候选目标(行人)，YOLO 模型输出其边界框的左上角与右下角位置 (x_1, y_1, x_2, y_2) 和置信度 Conf。

为了筛选出可靠的行人框，本文对每个行人边界框的置信度进行检测。只保留置信度 Conf 大于 0.5 的行人框，这样可以过滤掉不太可靠的检测结果。对于通过置信度筛选后的行人边界框位置 (x_1, y_1, x_2, y_2) ，提取行人关键点以代表其位置信息。为了提高计算效率，本文采用了行人边界框的中心点位置作为行人关键点。关键点的计算方法如下：

$$x_c = \frac{x_1 + x_2}{2} \quad (1)$$

$$y_c = \frac{y_1 + y_2}{2} \quad (2)$$

计算得到的 (x_c, y_c) 即为该行人的关键点，它代表了每个有效行人在关键区域中的位置。总之，通过 YOLO 模型的输出信息，本文首先筛选出置信度高于阈值的行人边界框。然后，通过计算边界框的中心点坐标，提取行人的关键点作为其位置的表示。这种行人关键点提取方法简单高效，为后续的行人分析任务提供了基础。

2.3. 行人社交距离估计

在行人数量较多的情况下，使用传统的两两组合方式计算行人之间的距离会导致算法计算量巨大。为了提高计算效率，本文引入了 KD-tree 数据结构[7]来构建行人的关键点坐标。KD-tree 是一种基于二叉树的数据结构，通过将空间划分为多个子空间，可以快速定位最近邻的行人。具体步骤如下：

步骤一：收集行人关键点。从行人检测结果中获取每个行人的中心点坐标。

步骤二：创建 KD-tree 数据结构。使用收集到的行人中心点坐标作为输入，构建一个 KD-tree 数据结构。该数据结构将中心点按照特定的划分规则进行排序和组织，以便快速搜索和定位。

步骤三：KD-tree 构建过程。在构建 KD-tree 时，首先选择一个维度作为划分依据。可以选择坐标轴上的某个维度，比如 x 轴或 y 轴。然后，根据选定的维度，找到中间位置的行人中心点，并将其作为当前节点。接下来，将剩余的行人中心点根据其在选定维度上的大小分配到左右子树中。这个过程递归进行，直到所有的中心点都被添加到 KD-tree 中。

步骤四：KD-tree 优化搜索。使用构建好的 KD-tree，可以进行快速的行人之间距离计算。对于给定

的两个行人中心点，通过从根节点开始的递归搜索，找到邻居行人。在搜索过程中，可以通过比较当前节点的划分维度上的值来确定搜索的方向，从而有效地减少搜索范围。

步骤五：距离估计。一旦搜索到 k 个邻居行人，可以计算其距离并进行风险评估。本文邻居行人数量设置为 5，这个参数可以根据实际需求调整。对于人流密集场合， k 取值较大；对于人流稀疏场合， k 取值较小。距离估计采用欧氏距离[8]公式：

$$d_{ij} = \sqrt{(x_c^i - x_c^j)^2 + (y_c^i - y_c^j)^2} \quad (3)$$

其中， d_{ij} 为第 i 个行人与 j 个行人的索引。 x_c^i 为第 i 个行人的 x 轴关键点位置， y_c^i 为第 i 个行人的 y 轴关键点位置。注意，这里的 j 是 i 行人在 KD-tree 搜索下的邻居行人。

步骤六：风险评估。根据预先设定的社交距离阈值，判断行人之间的距离是否在安全范围内。根据判断结果，进行相应的处理和预警操作。本文使用双阈值检测，当距离 d_{ij} 小于高危阈值 th_1 时，代表高危距离，那么会给予红色危险标识，同时给予标签“D”；如果是距离 d_{ij} 小于预警阈值 th_2 时，代表进入预警状态，给予紫色注意标识，同时给予标签“A”。其他距离，则标识为绿色安全，如图 3 所示。



Figure 3. Illustration of social distancing risk strategy
图 3. 社交距离风险策略可视化

2.4. 安全距离检测系统流程

下面给出里本文所提的关键区域人群社交距离检测方法的系统流程，如图 4 所示。核心流程如下：

步骤一：加载 YOLO 模型。将预训练的 YOLO 模型加载到计算机内存中，以便进行行人检测。

步骤二：图像输入。将待检测的图像输入到 YOLO 模型中，该图像可以从摄像头、视频文件或静态图像获取的[9]。

步骤三：YOLO 行人检测。通过 YOLO 模型对输入图像进行行人检测，得到每个检测到的行人的边界框(bounding box)和置信度分数。

步骤四：KD-tree 检测邻居行人。对于每个检测到的行人，使用 KD 树算法检测其邻居行人。KD 树是一种用于高效搜索最近邻居的数据结构，通过构建 KD 树并利用其搜索功能，可以快速找到每个行人周围的其他行人。

步骤四：估计邻居行人距离。通过测量每个行人与其邻居行人之间的距离，可以估计它们之间的相

对位置关系。这可以通过计算两个行人中心之间的欧氏距离或其他距离度量来实现。

步骤五：安全距离判断。根据行人之间的距离，进行安全距离判断以确定行人的状态。如果两个行人之间的距离小于设定的安全距离阈值，则被判定为危险状态或注意状态；否则，被判定为安全状态。

1) 标识分类：根据行人的状态，对其进行标识分类。危险状态可以用红色的危险标识表示，用于警示周围的观察者注意。安全状态可以用绿色的安全标识表示，表示行人处于安全状态。另外，还可以使用黄色的注意标识来表示行人处于一种中间状态，需要额外的关注。2) 可视化输出：根据行人的标识分类结果，将其可视化输出到图像上。可以使用不同颜色和标志符号来表示不同的标识分类，以便人们能够直观地理解和识别行人的状态。

循环步骤四和步骤五：遍历每个检测到的行人，以实现对整个图像中行人的检测和状态判断。

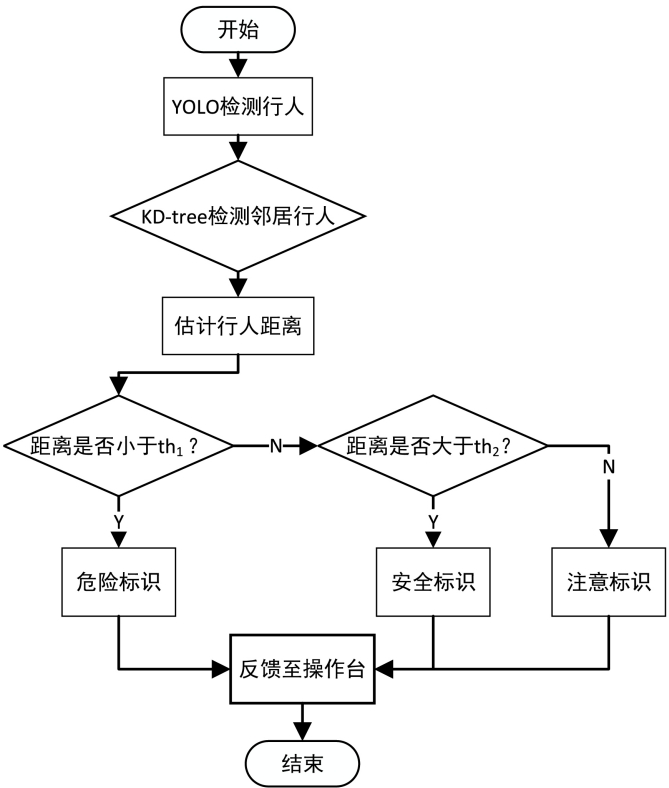


Figure 4. The workflow of the safety distance detection system
图 4. 安全距离检测系统流程

3. 运行结果与分析

3.1. 引言

本章节主要是对课题的实验结果进行分析，并归纳总结了系统环境的搭建，对每个功能模块的功能进行详细的介绍并测试，以便验证每个功能模块的实现情况，最后对课题的实验运行结果进行展开讨论。

3.2. 系统环境的搭建

实验采用如下的硬件配置：CPU 为 Intel(R) Core(TM) i9-13900，内存为 128GB，显卡为 NVIDIA GeForce RTX 3090。软件配置方面，采用 Ubuntu 操作系统，显卡驱动版本为 510.47.03，并且使用 CUDA

11.6 作为计算统一设备架构。深度学习框架选择 PyTorch 1.13, 标注工具采用 LabelImg, 编程语言为 Python 3.8。这样的硬件和软件配置为实验提供了稳定的计算环境, 以确保准确性和性能的实验结果。

3.3. 模型评价指标

在本节, 将使用已经标注为人的数据集对 YOLO 模型的参数进行微调训练。这些数据集包含了从网络 and 实际拍摄中获得的图像, 数据的来源是含有行人的数据集, 标注的类别是 person, 数据集的规模是 2000 张。

为了评估 YOLO 模型在目标检测方面的性能, 本文使用了多个评价指标, 包括检测精度(Precision)、召回率(Recall)、mAP@0.5 和 mAP@0.5:0.95。Precision 表示模型正确识别有目标判例占有所有模型认为有目标判例的比例。Recall 表示模型正确识别有目标判例占有所有正确判例的比例。mAP 是各类别的精确率-召回率曲线下的面积的平均值, 用于评估模型在所有类别上的性能。mAP@0.5 代表在 IOU 阈值为 0.5 时的平均 AP, 主要用于衡量模型的目标识别能力。mAP@0.5:0.95 表示在从 0.5 到 0.95 的多个 IOU 阈值范围内的平均 mAP, 主要用于评估模型的目标定位效果和边界回归能力。具体指标公式描述如下:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \quad (6)$$

$$\text{AP} = \frac{\sum P}{\text{Number}(\text{Total Objects})} \quad (7)$$

$$\text{mAP} = \frac{\text{AP}}{N(\text{class})} \quad (8)$$

其中, TP (True Positive)表示模型正确预测的正例数, 即正确分类为目标的本样本数量。FP (False Positive)表示模型错误地将负例预测为正例的数量, 即错误的预测结果。FN (False Negative)表示模型错误地将正例预测为负例的数量, 即漏掉的目标样本数。

3.4. 行人检测结果

本文使用的 YOLO 模型主要专注于图像中人物的目标识别任务, 并进行了 100 轮的训练。图 5 展示了 YOLO 模型的训练和评估结果, 其中横轴为网络迭代步数, 纵轴依次为 train/box_loss、train/obj_loss、metrics/precision、metrics/recall、val/box_loss、val/obj_loss、metrics/mAP_0.5、metrics/mAP_0.5:0.95。结果表明, 模型的训练过程收敛速度较快, 并最终趋于稳定状态。边界框和目标检测的训练和验证损失不断减少, 表明模型逐渐学习到了有效的目标识别策略。同时, 精确率、召回率以及 mAP@0.5 和 mAP@0.5:0.95 指标也逐渐提高, 说明模型在目标检测任务上的性能逐步提升。此外, 模型在训练过程中没有出现过拟合或欠拟合的情况, 这意味着模型能够很好地泛化到新的样本上, 并具有较高的鲁棒性和准确性。因此, 可以认为模型检测出的结果是准确可靠的, 可以在实际应用中进行行人目标的有效识别。

进一步, 本文利用 YOLO 模型对各种场景下的行人目标进行了测试, 并以直观的方式展示了模型的性能, 如图 6 所示。结果表明, 通过采用 YOLO 模型, 能够准确地检测出图像中的行人目标, 这表明该模型具备良好的泛化能力和鲁棒性。这为后续的安全距离检测提供了稳定可靠的基础。

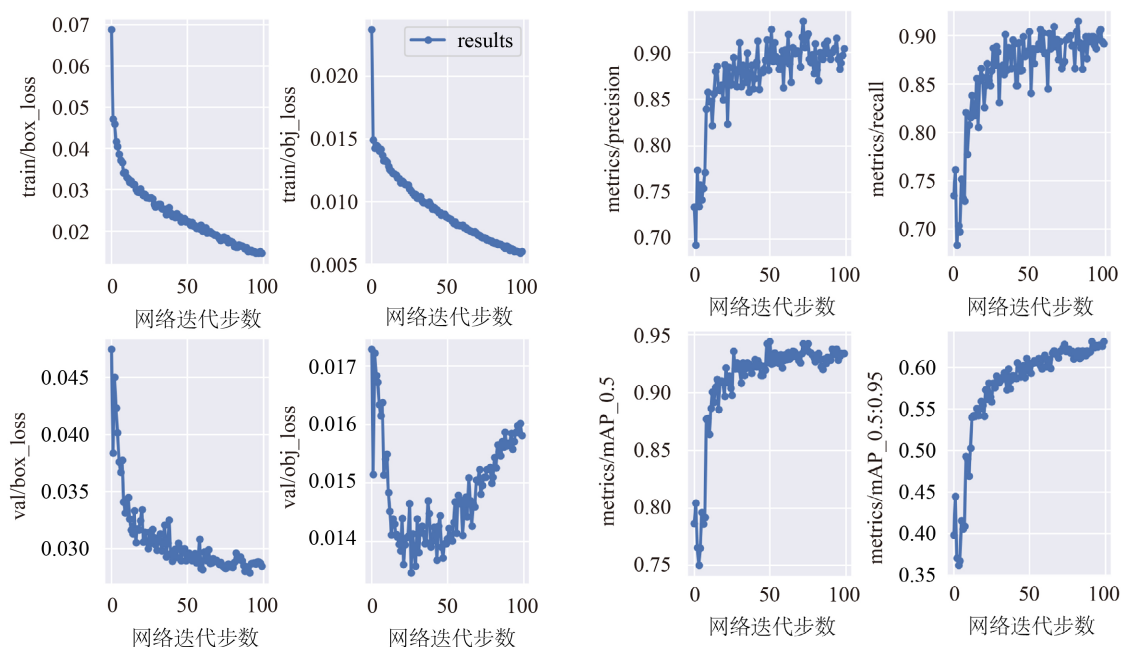


Figure 5. The variations of different metrics with training iterations

图 5. 各项指标随训练迭代变化曲线

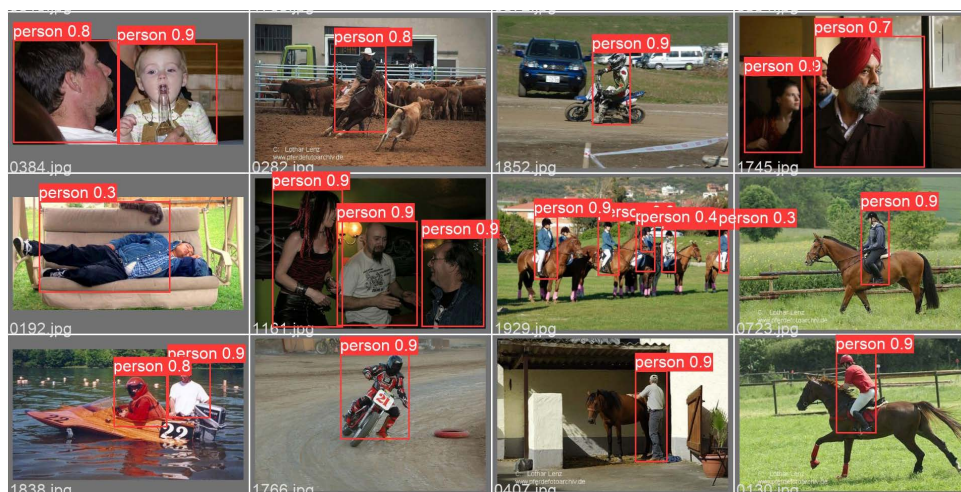


Figure 6. Illustration of person YOLO object detection

图 6. 人类 YOLO 目标检测可视化

3.5. 行人社交距离风险评估可视化

本节将可视化本文所提的行人社交距离。首先通过监控摄像头捕获公共区域实时图像视频；然后再将这些图像使用上节训练的 YOLO 模型进行行人目标检测；接着，对于置信度大于 0.5 的行人进行关键点提取；最后，根据行人的关键点使用 KD-tree 计算社交距离，并进行安全距离判断以确定行人的状态。如果两个行人之间的距离小于设定的危险距离阈值，则被判定为危险状态，标识为红色“D”；如果小于注意距离阈值，则被判断为注意预警状态，标识为紫色“A”；剩余情况，被判定为安全状态。

通过应用该方法于不同场景，如球场、街道、商场等，图 7 展示了行人安全社交距离的风险评估结果。结果表明，本文提出的方法能够较准确地估计行人之间的距离，并评估其安全社交距离风险。



Figure 7. Illustration of pedestrian social distancing risks
图 7. 行人社交距离可视化

4. 总结和展望

本论文提出了一种基于 YOLO 模型的人群社交距离检测方法,用于应对人流密集的公共区域的人流控制和预防疫情传播。该方法通过 YOLO 模型进行行人检测,获取行人的位置关键点信息。然后利用 KD-tree 算法检测行人之间的邻居关系,并估计它们之间的距离。根据设定的安全距离阈值,对行人的社交距离进行风险评估,包括危险、安全或需要注意的状态。实验结果验证了本文所提方法的有效性。

本文后续的工作,可以进一步改进该方法以适应更复杂的场景[10]和不同的人群行为。例如,考虑行人的方向性和速度变化,以更精确地评估社交距离。此外,可以结合其他技术,如图像增强和深度学习模型的优化,进一步提高检测的准确性和效率。此外,将该方法与其他安全措施相结合,如人脸识别和温度检测等,可以建立一个更完善的公共区域安全管理系统。

基金项目

国家级大学生创新创业训练项目(项目编号: 202213275005)。

参考文献

- [1] 王林, 张江涛. 基于轻量化神经网络的社交距离检测[J]. 计算机系统应用, 2023, 32(2): 128-138.

-
- [2] 王鑫鹏, 王晓强, 林浩, 李雷孝, 杨艳艳, 孟闯, 高静. 深度学习典型目标检测算法的改进综述[J]. 计算机工程与应用, 2022, 58(6): 42-57.
 - [3] 许德刚, 王露, 李凡. 深度学习的典型目标检测算法研究综述[J]. 计算机工程与应用, 2021, 57(8): 10-25.
 - [4] 王婧媛, 方健. 基于 Yolov5 的密集场所人数估计方法[J]. 吉林大学学报(信息科学版), 2021, 39(6): 682-687.
 - [5] 谈世磊, 别雄波, 卢功林, 谈小虎. 基于 YOLOv5 网络模型的人员口罩佩戴实时检测[J]. 激光杂志, 2021, 42(2): 147-150.
 - [6] 戴金波. 基于视觉信息的图像特征提取算法研究[D]: [博士学位论文]. 吉林: 吉林大学, 2013.
 - [7] Tao, S., Liao, H.Y. and Ganesh, S. (2022) Efficient Local Refinement near Parametric Boundaries Using kd-Tree Data Structure and Algebraic Level Sets. *Algorithms*, **15**, Article 245. <https://doi.org/10.3390/a15070245>
 - [8] 张闯, 王婷婷, 孙冬娇, 葛益娴, 常建华. 基于欧氏距离图的图像边缘检测[J]. 中国图象图形学报, 2013, 18(2): 176-183.
 - [9] Hao, Q., Xin, G. and F, Y. (2023) Fast Recognition Method for Multiple Apple Targets in Complex Occlusion Environment Based on Improved YOLOv5. *Journal of Sensors*, **2023**, Article ID: 3609541. <https://doi.org/10.1155/2023/3609541>
 - [10] Ranjai, B. and Heon, J. (2022) YOLOv5 with ConvMixer Prediction Heads for Precise Object Detection in Drone Imagery. *Sensors*, **22**, Article 8424. <https://doi.org/10.3390/s22218424>