基于检测的行人追踪方法研究

摘要

本文介绍了一种新型的基于检测的行人追踪方法。该方法采用YOLOv8进行行人检测，并结合神经网络对检测到的行人进行特征提取。通过贪心算法和匈牙利算法进行特征匹配，我们的方法能够有效地跟踪行人，即便在复杂的遮挡情况下也表现出良好的稳定性和准确性。

1. 引言

1.1 研究背景

行人追踪技术在视频监控、城市安全、自动驾驶车辆等领域中扮演着重要角色。随着计算机视觉和深度学习技术的飞速发展，行人追踪技术也取得了显著的进步。然而，由于行人遮挡、照明变化、姿态变化等因素，准确地追踪行人在视频中的运动仍然是一个挑战性的任务。

1.2 现有技术挑战

尽管传统的行人追踪方法，如基于区域的追踪和特征点追踪，已经在某些场景中取得了一定的成功，但它们通常难以处理复杂场景中的行人遮挡和快速运动问题。此外，这些方法在实时处理和准确性方面往往存在限制。近年来，深度学习方法因其强大的特征学习能力而在行人追踪领域获得广泛关注，但如何有效整合这些技术以提高追踪性能，仍然是一个开放的研究问题。

1.3 本研究的贡献

针对上述挑战，本研究提出了一种基于YOLOv8的行人检测和特征提取方法，结合神经网络和先进的匹配算法，以提高追踪的准确性和稳定性。我们的方法不仅能有效处理行人遮挡和光照变化等问题，而且能在实时处理的同时保持高准确度。本研究的主要贡献包括：

1. 利用YOLOv8进行高效的行人检测，提高检测精度和速度。
2. 应用深度学习技术对行人特征进行准确提取，增强追踪算法对个体的辨识能力。
3. 结合贪心算法和匈牙利算法优化特征匹配过程，提高追踪的连续性和稳定性。

2. 方法

2.1 特征提取

特征提取是行人追踪中的关键步骤，它决定了追踪算法能否准确识别和区分不同的行人。在本研究中，我们采用了两阶段的追踪方法，首先利用YOLOv8进行行人检测，然后使用不同方法提取行人的特征向量。

2.1.1 使用YOLOv8的行人检测的边界框

对于检测出的行人，其在相邻帧的位移量很小。这也就代表着，对于当前帧检测出的某一行人，只要其相比前几帧的一个行人位移量很小，我们有很大把握可以将其视做同一个人。因此在这种方案中，我们将yolov8检测出的行人的边界框作为该行人的特征向量。

2.1.2 利用神经网络提取特征向量

在这种方案里，我们着重研究每个被检测出来行人本身的纹理，颜色，尺度特征等。在行人被YOLOv8检测出之后，我们使用深度神经网络对每个行人进行特征提取。该网络旨在从每个行人中提取一个特征向量，该向量能够代表行人的唯一身份特征。

为了实现这一点，我们采用了基于卷积神经网络（CNN）的架构。这种架构能够从行人图像中提取出丰富的特征，包括但不限于行人的服装纹理、颜色、体态等。

以vgg19为例，我们采用pytorch里提供的预训练好的网络。因为其已经在ImageNet中进行了预训练，所以我们假定其可以很好的提取出不同行人之间独特的特征；同时，考虑到不同行人之间的特征主要是小尺度和中尺度之间的差异，因此选取的卷积层大多靠近网络输入层与中间层。

VGG19		
序号	层结构	
1	conv1-1	1
2	relu1-1	
3	conv1-2	2
4	relu1-2	
5	pool1	
6	conv2-1	3
7	relu2-1	
8	conv2-2	4
9	relu2-2	
10	pool2	
11	conv3-1	5
12	relu3-1	
13	conv3-2	6
14	relu3-2	
15	conv3-3	7
16	relu3-3	
17	conv3-4	8
18	relu3-4	
19	pool3	
20	conv4-1	9
21	relu4-1	
22	conv4-2	10
23	relu4-2	
24	conv4-3	11
25	relu4-3	
26	conv4-4	12
27	relu4-4	
28	pool4	
29	conv5-1	13
30	relu5-1	
31	conv5-2	14
32	relu5-2	
33	conv5-3	15
34	relu5-3	
35	conv5-4	16
36	relu5-4	
37	pool5	
38	fc6(4096)	17
39	relu6	
40	fc7(4096)	18
41	relu7	
42	fc8(1000)	19
43	prob(softmax)	

图1. 选取特征层

如图1所示，我们选取了vgg19第一，第二卷积块的第一个卷积层作为行人的小尺度的特征，第三，第四卷积块的第一个卷积层作为行人的中尺度特征，第四卷积块的最后一个卷积层和第五卷积块的第一个卷积层作为行人的大尺度特征。

为了进一步提升该网络在“特征提取”上的表现，我们特别微调了这个网络，网络被训练以区分不同行人的微妙特征，即使在行人服装或姿态发生变化的情况下也能保持高度的区分度，以提高其在行人重识别（Re-ID）任务中的表现。

训练数据来源MOT15训练集，根据标注数据将视频中的每一帧所有行人裁剪出来，构成自建的数据集；由于该任务没有合适的评价指标，因此没有设置测试集。因此我们简单的认为当损失函数收敛时训练完成。

微调过程如下：左侧的网络输入的数据是行人的原始图像，并且不进行梯度更新；而右侧的网络输入图像是左侧网络输入图像经25%~40%遮挡而形成的，我们希望网络提取到的遮挡行人的特征和他本人的特征越相近越好，因此损失函数采用MSE。如图2所示。

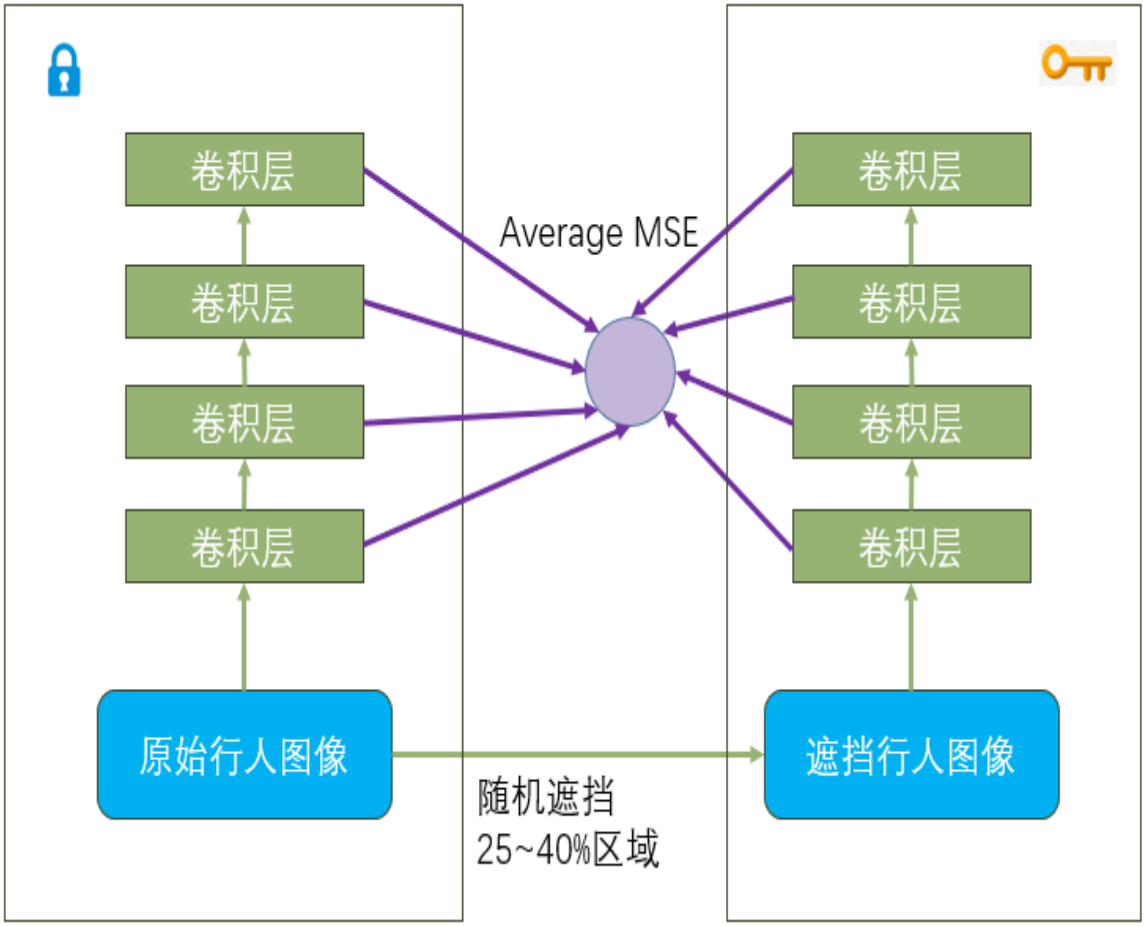

图2. 训练流程

图3. Loss下降曲线

2.2 特征匹配

2.2.1 特征向量与相似度的定义

在第一种特征提取方法中，每个行人的特征向量就是其边界框，不同行人的相似度是指不同行人之间边界框的交并比，我们利用交并比来衡量不同帧间行人的位移量

在第二种特征提取方法中，每个行人的特征向量就是网络输出的特征图，不同行人的相似度是指不同行人之间特征图余弦相似度的平均。

2.2.2 贪心算法

贪心算法是特征匹配中的一种简单而有效的方法。它基于每个行人特征向量之间的相似度进行匹配。算法计算当前帧中检测到的行人与前几帧中行人特征向量之间的相似度。然后对当前帧剩下的行人，再重复该步骤。若没有成功匹配，则认为是新出现的行人。从而实现行人的连续追踪。

此方法的优点在于其简单和速度快。在实时追踪应用中，这是非常重要的。然而，由于贪心算法做的是局部最优解，其可能在面对复杂场景时出现匹配错误，特别是在行人交叉或遮挡的情况下。

2.2.3 匈牙利算法

为了克服贪心算法的局限性，我们进一步采用了匈牙利算法，这是一种经典的优化算法，用于解决分配问题。在行人追踪的场景中，匈牙利算法用于在多个视频帧之间最优地分配行人标识。

具体而言，算法首先构建一个成本矩阵，其中每个元素代表当前帧中一个行人与上一帧中一个行人之间的特征相似度的函数，在这里，我们将其具体为：

$$\text{cost}[i][j] = 1 - \text{Similarity}_{ij}$$

然后，匈牙利算法寻找一个能够最小化总成本的分配方案。这种方法在保证每个行人只被匹配一次的同时，尽量减少匹配错误。

匈牙利算法提供了一种更为全面和精确的匹配策略，可以有效地提高追踪的准确性。

3. 改进

通过对实验视频的分析：

对于第一种特征提取方案，在一段时间内没被遮挡的人的匹配是鲁棒的。而出现遮挡会导致追踪目标短暂消失，因此在其再次出现时，IoU的值会低于阈值/或其不是最优解,导致被错误识别为新目标。

对于第二种特征提取方案，追踪被遮挡行人的效果是比较好的。但是由于提取到的不同行人特征相似度相差不大，导致其在长时间追踪同一人时，仍会有很高的错误率。

因此，最终的算法是将二者结合：

在特征提取上，主体仍基于边界框，先利用边界框进行第一轮匹配

对于因IoU过低而被刷掉的行人再利用网络提取其特征，与未被匹配到的前几帧的行人进行第二轮匹配

若经过两轮匹配后，当前帧仍有部分行人未得到匹配，则将其视作新出现的行人

4. 结论

对于最终的算法，其在低密度人群中追踪效果十分良好，在有遮挡的情况下基本也可达到95%以上的追踪正确率；但是在高密度人群中，其表现效果极差，这是由于作为特征提取器的网络部分不能有效的表征出行人的独特的特征，在行人数量过大时，总会出现若干个行人的特征向量十分相似，导致错误匹配。

此外，作为特征提取器的主体，网络的选择也是十分重要的：

对于vgg19而言，其可以较好地提取不同行人之间地特征，但是他的运行时间会比较长，难以做到实时的效果。

而对于轻量化的ResNet18而言，其追踪效率很高，几乎可以达到实时的效果，但其提取行人特征的能力不如vgg19，导致追踪的表现往往也差于vgg19。

所以该算法后续的主要提升方向仍是特征提取器的选择与构建，如何构建出运行速度更快，特征提取更准的网络仍是主要难点之一。

5. 附录

本文所有代码，数据，实验视频均在github仓库：https://github.com/Win-commit/Pedestrian_trackingMOT.git