Computer Science and Application 计算机科学与应用, 2021, 11(4), 783-794
Published Online April 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.114080

Hans汉斯

基于轻量级人体姿态估计和图卷积的摔倒实时检测方法

何炜婷，曾　碧，陈文轩

广东工业大学计算机学院，广东 广州
Email: heweiting95@foxmail.com

收稿日期：2021年3月9日；录用日期：2021年4月6日；发布日期：2021年4月13日

摘　要

基于人体姿态估计的摔倒检测方法，因其人体姿态估计模型涉及十几个关节点的识别与处理，导致整体模型的检测速度较慢。为了摔倒检测达到实时性，提出了一种基于轻量级人体姿态估计模型和图卷积的摔倒实时检测方法。该方法首先采用优化后的基于目标检测的两阶段轻量级人体姿态估计模型进行关节点检测，使整体模型达到轻量级；然后使用只有6个特征提取模块的时空图卷积网络对人体关节点序列进行摔倒检测，提高整体模型摔倒检测的准确率。本文通过NTU-D-RGB-120和UR Fall Detection Dataset两个数据集进行实验，摔倒检测的正确率达到96.1%，整体模型在GTX1060Ti显卡中达到约33FPS。

关键词

人体姿态估计，图卷积网络，轻量级，摔倒检测，动作识别

Real-Time Fall Detection Based on Lightweight Human Pose Estimation and Graph Convolution Network

Weiting He, Bi Zeng, Wenxuan Chen

School of Computers, Guangdong University of Technology, Guangzhou Guangdong
Email: heweiting95@foxmail.com

Received: Mar. 9th, 2021; accepted: Apr. 6th, 2021; published: Apr. 13th, 2021

文章引用: 何炜婷, 曾碧, 陈文轩. 基于轻量级人体姿态估计和图卷积的摔倒实时检测方法[J]. 计算机科学与应用, 2021, 11(4): 783-794. DOI: 10.12677/csa.2021.114080

Abstract

The fall detection based on human pose estimation, because the human pose estimation involves the recognition and processing of more than a dozen joint points, the detection speed of the overall model is slow. In order to achieve real-time fall detection, a real-time fall detection method based on a lightweight human pose estimation and graph convolution network is proposed. The method first uses an optimized two-stage lightweight human pose estimation based on object detection to detect joint points, so that the overall model is lightweight; then uses the spatio-temporal graph convolutional network with only 6 feature extraction modules to perform fall detection on the human joint point sequence to improve the accuracy of the overall model fall detection. This article conducts experiments on two data sets, NTU-D-RGB-120 and UR Fall Detection Dataset, and the accuracy rate of fall detection reaches 96.1%, and the overall model reaches about 33FPS in the GTX1060Ti.

Keywords

Human Pose Estimation, Graph Convolutional Network, Lightweight, Fall-Down Detection, Action Recognition

1. 引言

我国开始步入老龄化社会，老年人数量众多。老人因为身体机能下降而容易发生摔倒的情况，如果不及时发现并进行救护，可能会导致一些严重的后果。

目前，检测摔倒的技术主要有两种方式：通过随身穿戴设备的传感器检测摔倒和通过摄像头的视频检测摔倒。通过穿戴设备检测的方法，需要人随身带着设备，例如手机、手环、腰带等。通过这些设备上的加速度计、陀螺仪等传感器采集的重心变化数据运用机器学习算法进行摔倒检测[1]。该检测方法虽然简单有效，但对于一些老人来说，他们会忘记或者不喜欢佩戴这些设备。通过摄像头的视频图像检测的方法则不需要人随身携带设备，但因图像数据本身就相对较大，处理起来较慢，所以检测算法的计算量较大，检测速度也会较慢。如何优化关于视频图像摔倒检测方法的速度是非常值得研究的一个方向。

通过视频图像检测摔倒的方法有很多种，例如基于边缘轮廓[2] [3]、基于密集光流[4] [5]、基于人体姿态估计[6] [7]、基于三维时空卷积网络(C3D) [8] [9]等。其中，基于人体姿态估计的方法不止可以用于实现摔倒检测，还可以用于实现其他动作识别、动作分析、人机交互等算法，相对于其他方法来说扩展性更强。此外，该方法不需要考虑环境背景干扰的问题，检测准确率会比其他方法相对较高。但该方法因人体姿态估计算法本身的计算量较大，检测速度相比其他方法较慢，难以达到实时检测的效果，所以本文提出一种基于轻量级人体姿态估计的摔倒实时检测方法。

本文的主要贡献如下：

(1) 提出了一种基于目标检测的两阶段轻量级人体姿态估计模型来优化整体摔倒检测的速度；

(2) 通过消融实验对比分析该人体姿态估计模型各阶段的 Loss 与精度，验证该模型优化方法的有效

性；

(3) 采用规模缩小后的图卷积网络来提高模型速度和摔倒识别的准确率；

(4) 在 NTU-D-RGB-120 和 UR Fall Detection Dataset 这两个数据集上进行实验，实验结果表明本文方法的优化有效。

2. 相关工作

基于人体姿态估计的摔倒检测方法主要分两个检测阶段。第一个阶段是人体关节点检测，即人体姿态估计。第二个阶段就是根据检测出来的关节点进行摔倒检测，即分类问题。无论第二阶段使用哪种方法，因为人体姿态估计涉及十几个关节点的识别与处理，整个模型的计算量大部分在人体姿态估计算法这块，所以为了达到实时效果，检测方法的优化重点在于人体姿态估计模型的性能。

2.1. 人体姿态估计

人体姿态估计算法的设计思路主要有两种：自顶向下和自底向上。自顶向下的方法分为两阶段，首先通过目标检测算法检测出人体检测框，然后再从人体检测框内检测关节点，代表算法有 G-RMI [10]、AlphaPose [11]、CPN [12]等。自底向上的方法也是分为两阶段，首先检测出图像中所有的关节点，然后再把全部关节点按一定的策略组合成每个人的姿态，代表算法有 Openpose [13]、PifPaf [14]、DeeperCut [15]等。一般自底向上的方法会比自顶向下的方法快。因为自底向上的方法只需要一次性识别出图像中所有的关节点，而自顶向下的方法检测关节点的次数随着图像中人数的增加而增加，所以一般轻量级的人体姿态估计会采用自底向上的方法。例如，Intel 公司的 Osokin 在 OpenPose 的基础上，提出 OpenPose 的轻量级版本 Lightweight OpenPose [16]。Osokin 把 Openpose 的特征提取网络 VGG-19 换成 MobileNet，并把 5 个修正预测的精炼阶段减少到 1 个。虽然该模型的精度下降了，但模型的速度却提高了不少。除此之外，Sekii [17]也是使用自底向上的方法思路提出一种基于 YOLO [18]目标检测网格级别的轻量级人体姿态估计模型，把像素级别的热力图预测换成网格级别的目标检测来预测关节点，从而大幅度地提升关节点检测的速度。使用轻量级的人体姿态估计模型，因其关节点的检测精度降低，从而导致后续整个摔倒检测的判断结果会有一定的影响。

2.2. 摔倒检测

摔倒检测一般会使用长短期记忆网络(Long Short-Term Memory, LSTM) [19] [20]、支持向量机(Support Vector Machine, SVM) [2]、随机森林(Random Forest, RF)等方法，但这些方法不一定能学到区分关节点的一些运动特征，误判率较高，例如躺在床上、坐在地上、蹲下、摔倒等行为的区分。所以 Yan 等人[21]提出一种基于人体关节点的时空图卷积网络 ST-GCN 进行动作识别，该模型能更好地学习到一些隐藏的人体关节点运动的特征，泛化能力更强。所以本文方法中第二阶段的摔倒检测会采用比传统判别算法泛化能力更强的图卷积网络来进行摔倒判断，从而提高摔倒检测的准确率。

3. 方法设计

基于轻量级的人体姿态估计和图卷积的摔倒实时检测方法的整体算法流程如图 1 所示。首先把实时的视频提取关键帧，然后把提取的关键帧图片输入到人体姿态估计模型中进行关节点检测。同时，根据检测出来的关节点生成目标跟踪框来进行目标追踪。然后，把同个跟踪 ID 的人的带有时序的关节点坐标序列输入到时空图卷积网络中。时空图卷积网络利用每个动作带有时序的坐标序列的前后变化特征进行动作分类，从而进行摔倒检测警告。

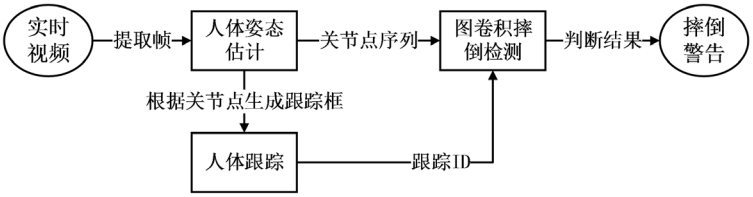

Figure 1. Flow chart of the overall algorithm of fall detection
图 1. 摔倒检测整体算法流程图

3.1. 轻量级人体姿态估计

为了整体算法模型能达到实时效果，需要将计算量较大的人体姿态估计模型进行优化，以达到实时检测的效果。本文参考文献[13] [16] [17]，提出一种基于目标检测的两阶段轻量级人体姿态估计模型Lightweight Pose Detection Network。如图 2 所示，该模型基于目标检测的思想，先把统一尺寸后的图像分为 $H \times W$ 个网格，然后用 CNN 网络来预测每个网格的候选的关节点和其连接的肢体，接着对候选框进行非极大值抑制(NMS)操作，最后利用 Hungarian Algorithm [22]算法生成每个人的姿态。

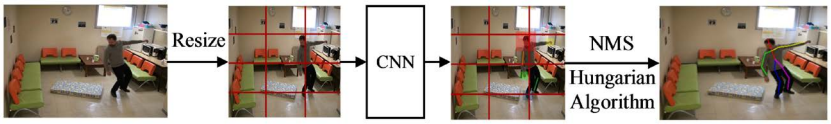

Figure 2. Flow chart of human pose estimation
图 2. 人姿态估计算法流程图

其中，CNN 网络有两个预测阶段，第一阶段网络主要预测细粒度的关节点，第二阶段网络是为了修正预测粗粒度的肢体连接，如图 3 所示。网络采用 MobilenetV2 [23]作为特征提取网络，然后经过两个 3 × 3 和一个 1 × 1 的卷积操作之后输出第一阶段局部特征的关节点预测结果与粗略的肢体连接预测结果。第二阶段为了修正属于全局特征的肢体连接预测，在特征网络输出的后面加入一层空洞卷积来增强感受野，并融合上一个阶段的关节点与肢体预测结果。特征融合之后，再经过两个 3 × 3 和一个 1 × 1 的卷积操作输出第二阶段修正的肢体预测结果。

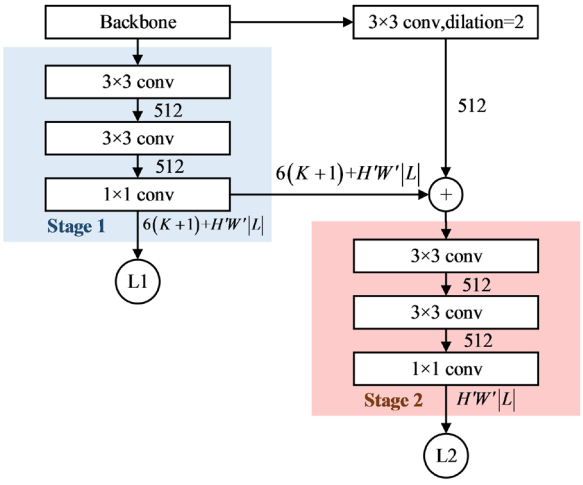

Figure 3. The network structure of the Lightweight Pose Detection Network
图 3. Lightweight Pose Detection Network 的网络结构

CNN 网络第一阶段的输出为 $6(K+1)+H'W'|L|$ 个通道。其中，$6(K+1)$ 为关节点预测信息，K 为要预测的关节点种类，6 为每个预测关节点所包含的信息数量。关节点预测包含的信息由公式(1)所示：

$$K_k^i=\left\{p\left(R|k,i\right),p\left(I|R,k,i\right),w_k^i,h_k^i,x_k^i,y_k^i\right\} \tag{1}$$

其中，$p\left(R|k,i\right)$ 为第 i 个网格预测关节点 k 的概率；$p\left(I|R,k,i\right)$ 为第 i 个网格预测关节点 k 的预测框与真实值的 IoU；x_k^i,y_k^i 为第 i 个网格预测关节点 k 的中心坐标相对于网格边界的距离；w_k^i,h_k^i 为第 i 个网格预测关节点 k 的预测框的宽和高，如图 4 所示。

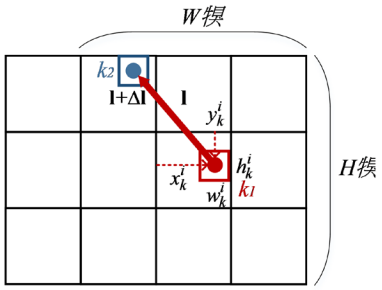

Figure 4. Schematic diagram of key points and limb prediction
图 4. 关节点与肢体预测示意图

其次，$H'W'|L|$ 为肢体预测的信息，其中，$|L|$ 为肢体的种类，$H'W'$ 为两个关节点连接的肢体范围，如图 4 所示。预测肢体连接的概率由公式(2)所示：

$$L_{k_1k_2}=\left\{p\left(L|k_1,k_2,\mathbf{l},\mathbf{l}+\Delta\mathbf{l}\right)\right\} \tag{2}$$

其中，$\Delta\mathbf{l}$ 的范围为 $H'W'$。

CNN 网络的第二阶段主要是为了修正粗粒度的肢体连接预测结果，所以网络输出的只有肢体预测的信息，输出通道为 $H'W'|L|$。

整体网络的损失函数为：

$$L_{\text{total}}=L1+L2 \tag{3}$$

$$\begin{aligned}L1=&\partial_{\text{resp}}\sum_{i\in\mathcal{9}}\sum_{k\in\kappa}\left\{\delta_k^i-\hat{p}\left(R|k,i\right)\right\}^2\\&+\partial_{\text{IoU}}\sum_{i\in\mathcal{9}}\sum_{k\in\kappa}\delta_k^i\left\{p\left(I|R,k,i\right)-\hat{p}\left(I|R,k,i\right)\right\}^2\\&+\partial_{\text{coor}}\sum_{i\in\mathcal{9}}\sum_{k\in\kappa}\delta_k^i\left\{\left(x_k^i-\hat{x}_k^i\right)^2+\left(y_k^i-\hat{y}_k^i\right)^2\right\}\\&+\partial_{\text{size}}\sum_{i\in\mathcal{9}}\sum_{k\in\kappa}\delta_k^i\left\{\left(\sqrt{w_k^i}-\sqrt{\hat{w}_k^i}\right)^2+\left(\sqrt{h_k^i}-\sqrt{\hat{h}_k^i}\right)^2\right\}\\&+\partial_{\text{limb1}}\sum_{i\in\mathcal{9}}\sum_{\Delta\mathbf{l}\in\iota}\sum_{(k_1,k_2)\in\chi}\max\left(\delta_{k_1}^i,\delta_{k_2}^j\right)\left\{\delta_{k_1}^i\delta_{k_2}^j-p\left(L|k_1,k_2,\mathbf{l},\mathbf{l}+\Delta\mathbf{l}\right)\right\}^2\end{aligned} \tag{4}$$

$$L2=\partial_{\text{limb2}}\sum_{i\in\mathcal{9}}\sum_{\Delta\mathbf{l}\in\iota}\sum_{(k_1,k_2)\in\chi}\max\left(\delta_{k_1}^i,\delta_{k_2}^j\right)\left\{\delta_{k_1}^i\delta_{k_2}^j-p\left(L|k_1,k_2,\mathbf{l},\mathbf{l}+\Delta\mathbf{l}\right)\right\}^2 \tag{5}$$

其中，$\delta_k^i\in\{1,0\}$ 表示该网格是否存在 k 关节点；$\left(\partial_{\text{resp}},\partial_{\text{IoU}},\partial_{\text{coor}},\partial_{\text{size}},\partial_{\text{limb1}},\partial_{\text{limb2}}\right)$ 为每个损失的权重。

最后 CNN 网络最终输出的预测结果只使用第一阶段的关节点预测信息和第二阶段的肢体预测信息。

CNN 网络输出关节点与肢体预测结果后，根据关节点的置信度 $p(R|k,i)\,p(I|R,k,i)$ 和肢体连接的置信度 $p(L|k_1,k_2,\mathbf{l},\mathbf{l}+\Delta\mathbf{l})$，通过匈牙利算法(Hungarian Algorithm) [22]用最大权重二分匹配来进行关节点连接，从而生成每个人的姿态，如图 5 所示。

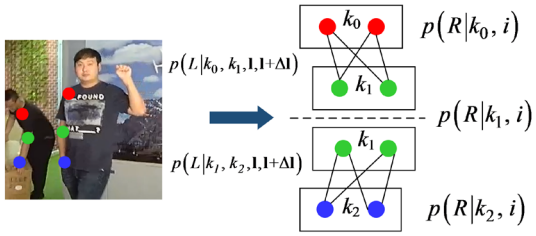

Figure 5. Schematic diagram of connection join points
图 5. 关节点连接示意图

3.2. 目标跟踪

虽然人体姿态估计采用的是自底向上的方法，没有生成人体检测框，但可以根据检测出来的关节点来生成目标追踪的检测框。遍历每个人的关节点坐标，找出最左、最右、最上和最下的坐标生成检测框。得出检测框后，用轻量级的 Sort 目标跟踪算法进行跟踪。Sort 算法[24]利用上下帧中的框重合度 IoU 和框内关节点的距离作为评判是否为同一个 ID 的标准，进而存储不同 ID 的连续关节点序列。因本文的人体姿态估计模型速度约 35 FPS，为了整体模型达到实时检测效果，设定暂存的阈值为 30 帧。然后使用卡尔曼滤波(Kalman filtering)对关节点形成的检测框以及运动规则的预测进行位置的评估优化，为生成下一帧追踪的特征提供更好的条件，从而形成稳定的追踪。最后，利用匈牙利算法(Hungarian Algorithm) [22]将追踪到的 ID 对图像中的关节点进行分配，分别存储帧中提取的不同的关节点数据。

3.3. 图神经网络摔倒识别

时空图卷积网络通过学习摔倒动作的带有时序的关节点坐标序列的前后变化特征来对摔倒进行分类识别。参考 ST-GCN [21]模型，把每帧全部关节点作为数据流，输入到模拟关节点沿空间和时间维度的结构化信息的时空图神经网络中，图 6 为所构建的时空关节点图数据流示意图，橙色线连接为空间维度，蓝色线连接为时间维度。

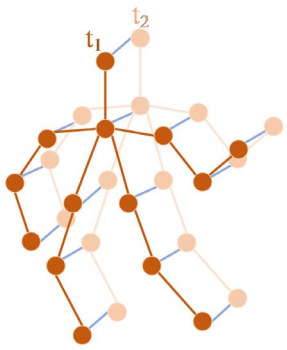

Figure 6. Spatio-temporal joint point graph data flow
图 6. 时空关节点图数据流

网络结构如图 7 所示。首先，摔倒动作的特征对比其他动作会相对较简单，所以仅采用 30 帧的连续

关节点数据作为网络的输入并对其位置特征进行归一化。然后，相应地把模型规模缩小，特征提取模块只使用 6 层：前 2 层为低维特征 64 通道数；中间 2 层为 128 通道数；后 2 层为高维特征 256 通道数。在特征模块中，ATT 是注意力模块，负责针对不同层的 GCN 提取的语义特征的作用，GCN 负责学习空间中相邻关节的特征，TCN 负责学习时间维度中关节点变化的特征。最后，输出部分进行池化后使用 FC 全连接层输出结果。

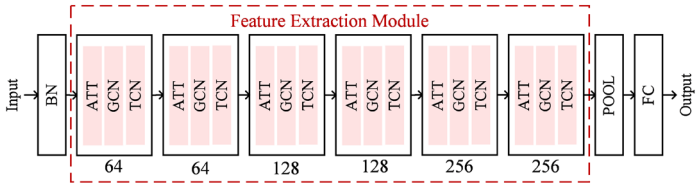

Figure 7. Spatio-temporal graph convolutional network
图 7. 时空图卷积网络结构

4. 实验

本文实验环境使用 Intel Core i7-6700 3.4 GHz 处理器与 GTX1060Ti 6 GB 独立显卡的笔记本电脑作为测试设备，使用 GTX1080Ti 11G 独立显卡作为训练设备。

4.1. 实验数据与处理

人体姿态估计模型因为较轻量级，所以数据集采用 MPII 数据集。MPII 数据集一共包含 4×10^4 个人，25000 张图片。使用官方划分训练与验证数据集。

摔倒识别的图神经网络使用 NTU-D-RGB-120 和 UR Fall Detection Dataset 数据集。NTU-D-RGB-120 数据集中截取 A9 standing up、A8 sitting down、A43 falling、A59 walking towards each other 和 A60 walking apart from each other 这 5 个动作分类样本作为数据集，在训练时需要手动筛选并分成多个 30 帧的可用数据集，去除 Z 坐标并换成 1 置信度。UR Fall Detection Dataset 数据集中截取 standing up 、sitdown、falling、waliking 、standing、sitting、lying 这 7 个动作分类样本作为数据集。其中，UR Fall Detection Dataset 数据集是没有关节点的标签，所以使用精度较高的 AlphaPose 人体姿态估计模型来输出制作视频数据集里的关节点标签。另外，由于使用的是二维的人体关节点，头部、肩部、臀部和腿部的运动特征对比其他关节点更能作为摔倒的判断依据，因此在数据预处理中需要筛选这些关节点训练的置信度，直接置为 1。过滤掉一些无效的视频数据后，整体图神经网络的数据集一共有 2300 个视频，其中训练集有 1600 个视频，验证集 700 个视频，每个类别平均分布。

4.2. 人体姿态估计分析

人体姿态估计模型训练输入的图片尺寸为 384 × 384，批次大小为 32，初始学习率设置为 5×10^{-4}，训练 150 轮，使用 SDG 随机梯度下降，冲量为 0.9，权重衰减 5×10^{-4}。损失函数的 loss 权重设置为。采用迭代训练的方式进行训练，先用 COCO 数据集训练特征提取网络，然后加上第一阶段网络用 MPII 数据集继续训练，最后加上第二阶段网络用 MPII 数据集一起训练。训练完成后，模型与其他模型的精度与速度的对比，如图 8 和表 1 所示。

为验证第一个阶段输出关节点和肢体连接预测与第二个阶段输出肢体连接预测的模型优化方法是有效的，设计消融实验进行验证。消融实验为使用第一阶段直接输出关节点与肢体预测结果与第二阶段直接输出关节点与肢体预测作对比，使用训练了 150 轮之后损失函数中输出的 L_{resp}、L_{IoU}、L_{coor}、L_{size}、L_{limb} 与 mAP 作为评价指标，如表 2 所示。

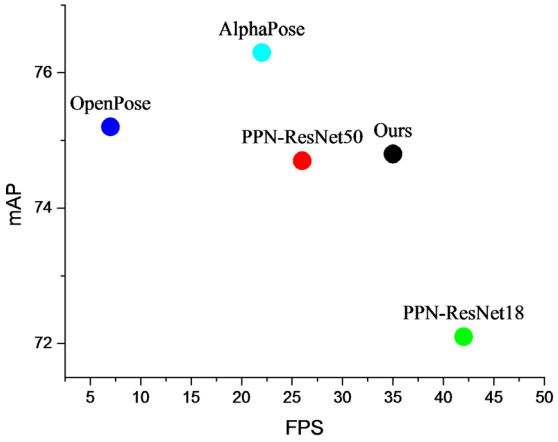

Figure 8. Comparison of accuracy and speed of different algorithms
图 8. 不同算法精度与速度对比

Table 1. Comparison of accuracy and speed of different algorithms on the MPII data set
表 1. MPII 数据集上不同算法精度与速度对比

	头部	肩部	手肘	手腕	臀部	膝盖	脚踝	mAP	FPS
AlphaPose [11]	87.2	85.8	77.4	69.7	74.0	72.2	64.4	76.3	22
OpenPose [13]	90.8	87.2	77.3	66.5	74.8	67.6	61.2	75.2	7
Pose Proposal Networks-ResNet18 [17]	92.5	88.5	73.8	62.1	71.7	61.9	54.6	72.1	42
Pose Proposal Networks-ResNet50 [17]	92.9	89.6	77.3	67.2	73.5	66.8	58.4	74.7	26
本文模型	93.1	89.5	77.3	67.4	73.4	66.9	58.7	74.8	35

Table 2. Analysis of ablation experiments
表 2. 消融实验分析

	L_{resp}	L_{IoU}	L_{coor}	L_{size}	L_{limb}	mAP	FPS
第一阶段	12.647	0.062	1.675	0.071	13.287	71.7	48
第二阶段	13.012	0.063	1.769	0.137	12.979	74.2	30
本文模型	12.654	0.062	1.676	0.073	12.982	74.8	35

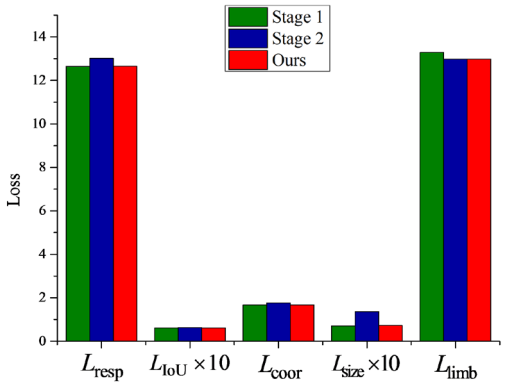

Figure 9. Loss comparative analysis of ablation experiments
图 9. 消融实验的 Loss 对比分析

从图 9 中，通过 Loss 损失函数的分析可以看出，第一阶段关节点相关的 Loss 比第二阶段小，细粒度的关节点预测更为准确；第二阶段肢体预测的 Loss 比第一阶段小，粗粒度的肢体预测更为准确。其次，从图 10 中，从两阶段网络模型输出的检测效果可以看出来，第一阶段的关节点预测会比第二阶段的更准确，第二阶段对肢体的预测会比第一阶段更加准确，所以验证了本文模型的优化方法有效。图 11 为优化后的人体姿态估计模型的检测效果。

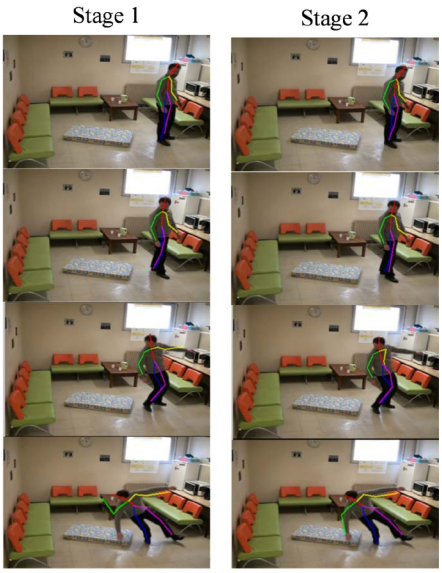

Figure 10. The detection results of stage 1 and stage 2
图 10. 第一和第二阶段检测效果

Figure 11. The detection results of the human pose estimatio
图 11. 本文人体姿态估计模型的检测效果

4.3. 摔倒识别分析

摔倒识别的时空图卷积网络先使用 NTU 数据集数据进行预训练，然后使用 UR 数据集进行进一步训练，训练使用 30 帧连续的 16 个关节点数据，批大小为 128，初始学习率为 1×10^{-3}，训练 50 轮，每 10 个 epoch 衰减 0.1，使用交叉熵损失函数以及 Adam 优化器。

本文采用 4 个评价指标，分别为精确度(Precision)、召回率(Recall)、准确率(Accuracy)和 F1-measure 综合性评价指标。首先，将实验样本分为四类：真正例 TP、假正例 FP、假反例 FN 和真反例 TN。真正

例 TP(True Position)表示正样本被正确地分类为正样本；假正例 FP (False Position)表示负样本被错误地分类为正样本；假反例 FN (False Negative)表示正样本被错误地分类为负样本；真反例 TN (TrueNegative)表示负样本被正确地分类为负样本。

所以，精确度(Precision)表示正确地分类的摔倒样本占分类为摔倒样本的比例，如公式(6)所示。

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

召回率(Recall)表示模型正确地分类的摔倒样本占实际摔倒样本的比例，如公式(7)所示。

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

准确率(Accuracy)表示模型分类地正确(包含摔倒和非摔倒)的样本占所有样本的比例，如公式(8)所示。

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \tag{8}$$

F1-measure 为分类常用的一个综合性评价指标，如公式(9)所示。

$$\text{F1-measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{9}$$

动作识别算法一般会使用 300 帧连续的关节点数据，但本文的人体姿态估计模型速度约 35 FPS，为了整体模型达到实时检测效果，设置使用 30 帧的数据。实验结果如表 3 所示，设置使用 30 帧，整体模型的摔倒检测准确率并没有下降多少。其次，把网络的特征提模块改成 6 层，摔倒检测的准确率也没有下降多少。

Table 3. Analysis of the influence of the number of key frames and the number of feature extraction layers on detection
表 3. 关键帧数和特征提取层数的设置对检测的影响分析

帧数	特征提取层数	Precision	Recall	Accuracy	F1-measure
300	9	0.973	0.947	0.981	0.961
30	9	0.970	0.942	0.975	0.956
30	6	0.957	0.925	0.961	0.941

虽然使用了轻量级的人体姿态估计模型，但使用时空图卷积网络进行摔倒识别之后，整体模型的准确率并没有比使用精度较高的 Alphapose 人体姿态估计模型的准确率低多少。此外，对比使用 SVM 动作分类的方法，准确率却要高很多，如表 4 所示。本文方法的摔倒检测效果如图 12 所示。

Table 4. Comparison of accuracy and speed of fall detection
表 4. 摔倒检测的准确率与速度对比

	Precision	Recall	Accuracy	F1-measure	FPS
SVM	0.797	0.237	0.801	0.343	36
Alphapose + GCN	0.963	0.932	0.969	0.949	23
本文方法	0.957	0.925	0.961	0.941	33

Figure 12. detection results of the fall detection
图 12. 摔倒检测效果

5. 总结

为了使基于人体姿态估计的摔倒检测方法达到实时检测的效果，本文先把计算量较大的人体姿态估计模型进行优化，提出一种基于目标检测的两阶段轻量级人体姿态估计模型。同时，为了验证两阶段网络的优化方法有效，设计实验分析各阶段网络预测关节点和肢体的效果。然后，为了提高整体检测方法的准确率与检测速度，使用规模缩小后的时空图卷积网络来对关节点序列进行摔倒检测。经过实验，本文方法对比使用高精度的人体姿态估计模型，整体算法的摔倒检测准确率并没有下降很多，但速度却提升了不少。

参考文献

[1] Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P. and Vecchio, A. (2012) A Smartphone-Based Fall Detection System. *Pervasive & Mobile Computing*, **8**, 883-899. https://doi.org/10.1016/j.pmcj.2012.08.003

[2] Feng, W., Liu, R. and Zhu, M. (2014) Fall Detection for Elderly Person Care in a Vision-Based Home Surveillance Environment Using a Monocular Camera. *Signal Image & Video Processing*, **8**, 1129-1138. https://doi.org/10.1007/s11760-014-0645-4

[3] Alhimale, L., Zedan, H. and Al-Bayatti, A. (2014) The Implementation of an Intelligent and Video-Based Fall Detection System Using a Neural Network. *Applied Soft Computing*, **18**, 59-69. https://doi.org/10.1016/j.asoc.2014.01.024

[4] Nunez-Marcos, A., Azkune, G. and Arganda-Carreras, I. (2018) Vision-Based Fall Detection with Convolutional Neural Networks. *Wireless Communications & Mobile Computing*, **2017**, Article ID: 9474806. https://doi.org/10.1155/2017/9474806

[5] Wang, H. and Schmid, C. (2013) Action Recognition with Improved Trajectories. *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, 1-8 December 2013, 3551-3558. https://doi.org/10.1109/ICCV.2013.441

[6] Chen, W., Jiang, Z., Guo, H. and Ni, X. (2020) Fall Detection Based on Key Points of Human-Skeleton Using Open Pose. *Symmetry*, **12**, 744. https://doi.org/10.3390/sym12050744

[7] Song, S., Lan, C., Xing, J., Zeng, W. and Liu, J. (2017) An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, **31**, 501.

[8] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M. (2015) Learning Spatiotemporal Features with 3d Convolutional Networks. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 4489-4497. https://doi.org/10.1109/ICCV.2015.510

[9] Xu, H., Das, A. and Saenko, K. (2017) R-c3d: Region Convolutional 3d Network for Temporal Activity Detection. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 5783-5792. https://doi.org/10.1109/ICCV.2017.617

[10] Papandreou, G., *et al.* (2017) Towards Accurate Multi-Person Pose Estimation in the Wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 4903-4911. https://doi.org/10.1109/CVPR.2017.395

[11] Fang, H.-S., Xie, S., Tai, Y.-W. and Lu, C. (2017) Rmpe: Regional Multi-Person Pose Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 2334-2343.

https://doi.org/10.1109/ICCV.2017.256

[12] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G. and Sun, J. (2018) Cascaded Pyramid Network for Multi-Person Pose Estimation. *Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 7103-7112. https://doi.org/10.1109/CVPR.2018.00742

[13] Cao, Z., Simon, T., Wei, S. and Sheikh, Y. (2017) Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 1302-1310. https://doi.org/10.1109/CVPR.2017.143

[14] Kreiss, S., Bertoni, L. and Alahi, A. (2019) PifPaf: Composite Fields for Human Pose Estimation. *Computer Vision and Pattern Recognition*, Long Beach, 16-20 June 2019, 11977-11986. https://doi.org/10.1109/CVPR.2019.01225

[15] Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M. and Schiele, B. (2016) DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. *European Conference on Computer Vision*, Amsterdam, 11-14 October 2016, 34-50. https://doi.org/10.1007/978-3-319-46466-4_3

[16] Osokin, D. (2019) Real-Time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. *International Conference on Pattern Recognition Applications and Methods*, Prague, 19-21 February 2019, 744-748. https://doi.org/10.5220/0007555407440748

[17] Sekii, T. (2018) Pose Proposal Networks. *European Conference on Computer Vision*, Munich, 8-14 September 2018, 350-366. https://doi.org/10.1007/978-3-030-01261-8_21

[18] Redmon, J., Divvala, S.K., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[19] Lin, H.Y., Hsueh, Y.L. and Lie, W.N. (2017) Abnormal Event Detection Using Microsoft Kinect in a Smart Home. 2016 *International Computer Symposium* (*ICS*), Chiayi, 15-17 December 2016, 285-289. https://doi.org/10.1109/ICS.2016.0064

[20] Lie, W.N., Le, A.T. and Lin, G.H. (2018) Human Fall-Down Event Detection Based on 2D Skeletons and Deep Learning Approach. 2018 *International Workshop on Advanced Image Technology* (*IWAIT*), Chiang Mai, 7-10 January 2018, 1-4. https://doi.org/10.1109/IWAIT.2018.8369778

[21] Yan, S., Xiong, Y. and Lin, D. (2018) Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition.

[22] Kuhn, H.W. (1955) The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, **2**, 83-97. https://doi.org/10.1002/nav.3800020109

[23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[24] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B. (2016) Simple Online and Real-Time Tracking. 2016 *IEEE International Conference on Image Processing* (*ICIP*), Phoenix, 25-28 September 2016, 3464-3468. https://doi.org/10.1109/ICIP.2016.7533003