A Lightweight Skeleton-Based 3D-CNN for Real-Time Fall Detection and Action Recognition

Nadhira Noor and In Kyu Park
Department of Electrical and Computer Engineering, Inha University
Incheon 22212, Korea
{nadhirannoor@gmail.com, pik@inha.ac.kr}

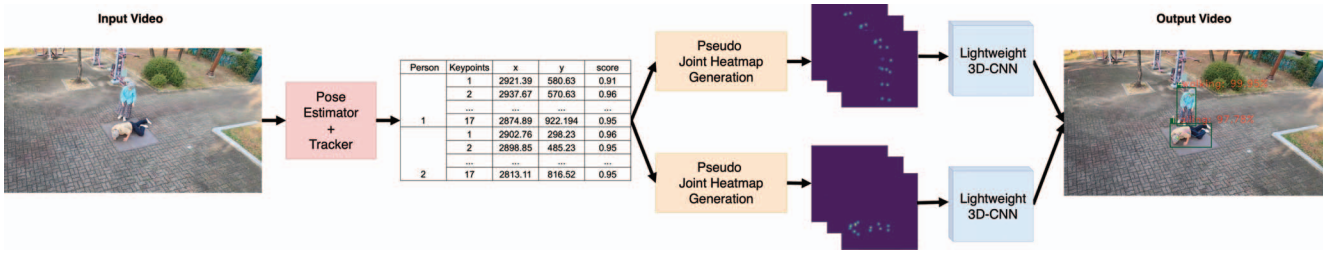

Figure 1: Overall framework of the lightweight skeleton-based 3D-CNN for real-time action recognition. For each frame in the video, we utilize a one-stage pose estimator to obtain the 2D keypoints. We then employ a tracking mechanism to establish the identity of each person, and save it as coordinate triplets. For every tracked identity, we generate the joint heatmap and stack it as input for our 3D-CNN model, which classifies the action performed by each tracked identity.

Abstract

Implementing skeleton-based action recognition in real-world applications is a difficult task, because it involves multiple modules such as person detection and pose estimaton. In terms of context, skeleton-based approach has the strong advantage of robustness in understanding actual human actions. However, for most real-world videos in the standard benchmark datasets, human poses are not easy to detect, (i.e. only partially visible or occluded by other objects), and existing pose estimators mostly fail to detect the person during the falling motion. Thus, we propose a newly augmented human pose dataset to improve the accuracy of pose extraction. Furthermore, we propose a lightweight skeleton-based 3D-CNN action recognition network that shows significant improvement on accuracy and processing time over the baseline. Experimental results show that the proposed skeleton-based method shows high accuracy and efficiency in real world scenarios.

1. Introduction

The latest research developments in deep learning action recognition are showing incredibly impressive performance. However, its practical implementation in real-world applications is challenging due to its computational heaviness. Deep learning models for action recognition often require significant computational resources, which makes them impractical for deployment on resource-constrained devices or in real-time scenarios. This limitation hinders the widespread adoption of deep learning-based action recognition systems in practical settings. In this paper, we explore techniques such as efficient architectures, input representations, and model compression to develop lightweight models that maintain high performance while reducing computational requirements. By reducing the computational cost, the practical usability of deep learning action recognition systems can be greatly enhanced, which enables their seamless integration into various real-world applications.

Moreover, in real-world applications, falling motion [2] has increasingly drawn healthcare industry demands for automated vision systems among types of human actions. Conventional fall detection methods typically rely on wearable sensors like smartwatches, necklaces, and smartphones. These devices employ 3D accelerometers to identify deviations in height orientation and velocity, which facilitates the detection of abnormal patterns associated with falls. However, the attachment of these devices to individuals' bodies is impractical. Alternatively, as the performance of vision-based action recognition has improved, it has be-

come feasible to detect falling motion directly from video footages. In particular, vision-based systems that utilize 2D or 3D skeletons as a simplified yet efficient representations of human poses.

In this paper, our approach addresses the issue of heavy computational requirements in deep learning action recognition. Figure 1 shows our overall framework. Specifically, we introduce a lightweight skeleton-based 3D-CNN action recognition network that is specifically designed for real-world applications. By leveraging the power of 3D convolutional neural networks (CNNs) and the simplicity of skeleton-based representations, our proposed pipeline achieves high accuracy in action recognition while significantly reducing the computational burden.

The core idea behind the skeleton-based approach is to leverage sequences of human poses to identify human actions or motions. By utilizing this sequence of poses, the method aims to accurately discern various human activities. However, prior works fail to recognize individuals during falling motion, particularly in highly occluded environments. This issue arises primarily due to the lack of falling motion examples in the human pose estimation dataset, which hampers the ability of the model to predict persons accurately in such dynamic scenarios. Consequently, the skeleton-based approach action recognition model struggles to identify individuals in falling motion. To address this challenge, we propose a human dataset as a solution. This dataset aims to enhance the performance of the person detection model by providing relevant falling motion examples, which enables the model to better handle dynamic scenarios and improve person recognition during falling motion. Moreover, we develop a real-time fall detection and action recognition method that effectively incorporates the proposed fall person detection dataset and utilizes our lightweight 3D-CNN action recognition model for the robust and timely detection of fall events and activity monitoring. The lightweight model efficiently processes the extracted pose keypoints, and the integration of the lightweight action recognition model further optimizes the performance of the system. This way ensures efficient and accurate identification of human actions without compromising accuracy. The contribution of this paper is summarized as follows.

- Newly augmented dataset for human pose, which improves the accuracy of pose estimation and thus results in more accurate action recognition
- Lightweight action recognition architecture that runs in real time in a multiperson environment
- State-of-the-art performance in fall detection

2. Related Works

2.1. Skeleton-based Action Recognition

Skeleton-based approaches heavily depend on pose extraction. Action recognition is considered as a time series problem; thus, early works utilized RNNs [10, 23]. In [10], the input is represented as 3D poses and is divided into five parts, while [23] directly uses joint locations of 3D poses as input. Another stream of works uses CNN [9, 26] models, which directly transform the coordinates in a skeleton sequence into a pseudoimage, typically a 2D input of shape $K \times T$, where K represents the number of joints and T is the temporal length.

Although RNN and CNN utilized joint coordinates, they did not explicitly exploit the structural topology of the joints; thus, Graph Convolution Network [35, 20, 24, 8] was introduced. GCN models such as ST-GCN [35] utilized the extracted keypoint sequences as spatio-temporal graphs. AS-GCN [20] employs multi-scale modeling and additionally predicts the human pose. G3D [24] proposed a novel graph convolution operator for capturing long range joint relationship modeling. Shift-GCN [8] aimed to reduce computational cost by proposing a shift graph operation.

However, in terms of multi-person, all GCNs method are very heavy because they multiply the input for each person detected. Therefore, a new approach using 3D-CNN, called PoseConv3D [11], was adopted to address multi-person scenarios without incurring additional computation costs. The 3D-CNN for action recognition was initially applied to RGB input, such as in SlowFast [13]. The SlowFast [13] network captured spatial semantics and motion separately by applying different spatio-temporal resolutions for the two networks. However, the use of RGB input demands a high number of channels, which makes the network computationally heavy. In contrast, PoseConv3D [11] offers a solution by representing the RGB input as stacks of joint heatmaps generated from extracted 2D human poses, which form 3D heatmap volumes as the input for 3D-CNN. This approach efficiently handles multi-person scenarios without burdening the network with excessive computation.

Furthermore, unlike 2D-CNN, 3D-CNN has the capability to learn spatio-temporal features, as a result, skeleton-based approaches using 3D-CNN outperforms GCN-based approaches, which makes them a more effective and efficient choice for action recognition tasks.

2.2. Vision-based Fall Detection

Vision-based fall detection can be categorized into two main approaches: RGB-based and skeleton-based. In early stages, RGB-based methods [7, 14, 15] learned human motion features by segmenting the subject from the background. This step helps isolate the human figure from the background. After this initial step, the system then ana-

lyzes the position and movements of the person to determine whether a fall has occurred or not. In some works alternative techniques are employed to represent the input video, for instance, [12] converts it into multiple dynamic images, while [27] transforms RGB input into optical flow images. Lu *et al*. [25] utilized 3D-CNN to extract 3D feature cubes. The input to the 3D-CNN is an image cube composed of 16 frames segmented from the video sequence, and the output is then used as input for LSTM models to classify the motion.

On the other hand, skeleton-based fall detection, as demonstrated in various studies [17, 21, 28], involves extracting 2D keypoints using a pose estimator model. These keypoints are then used as input for the system. A commonly employed technique in this approach is to utilize LSTM for handling the extracted keypoints sequences. Apicella *et al*. [4] incorporate additional techniques to enhance accuracy. They used an additional CNN to generate sequences of keypoints that the pose estimator failed to extract, which further improved the overall accuracy.

2.3. Human Pose Dataset

Multi-person pose estimation has achieved significant progress in the past few years, and a few popular benchmark datasets are available. Most of these datasets [22, 3, 19] annotate body keypoints for images in the wild. PoseTrack [6] provides dense annotations of video sequences with 15 body keypoints. Among these datasets, COCO [22] has emerged as one of the most widely used and popular resource for body keypoint localization. It offers comprehensive annotations of 17 keypoints for human bodies in challenging and uncontrolled conditions. These datasets collectively play a crucial role in advancing the state-of-the-art methods in multi-person pose estimation and continue to foster the development of robust and accurate pose estimation models.

3. Augmented Human Pose Dataset

In our work, we propose an augmented human pose dataset by building upon an existing dataset, namely the AI Hub senior abnormal behavior dataset [1], we exclusively utilize videos depicting falling motion. Our approach involves several key steps to enhance suitability of the dataset for our specific objective.

To begin, we shorten the video duration from 3-10 minutes to 3-7 seconds, by focusing solely on capturing and highlighting falling motions. This adjustment is made based on the labels of the dataset, which allows us to extract relevant segments effectively.

Next, we downsample the frame rate, which enables clearer identification of pose differences between frames. Although the original videos have a frame rate of 25 frames per second (FPS), through empirical analysis, we determine that sampling every 12^{th} frame satisfactorily captures typical falling motions. We manually annotate the bounding box around the person in each frame to facilitate the subsequent steps. This annotation is performed using the labeling tool, LabelImg [32], which ensures accurate delineation of the person's region of interest. The frames, along with their corresponding bounding box information, are then inputted into a 2D pose estimator to produce estimated keypoints. These estimated keypoints serve as the pseudo ground-truth for our augmented dataset.

Finally, we combine our augmented dataset with the COCO-pose [22] dataset, which results in a comprehensive dataset comprising a total of 69,279 person images and 2D keypoints information. By merging these datasets, we enhance the diversity and richness of the human pose dataset, which ensures its suitability for robust human pose estimation tasks for our action recognition model.

4. Proposed Method

In this section, we present the proposed method, which aims to establish an efficient pipeline for action recognition in real-world applications.

4.1. Overall Framework

Figure 1 shows our overall framework with two main modules: pose estimation and action recognition. In this section, we discuss both modules and the process of translating 2D keypoints into joint heatmaps, which serve as the input to our 3D-CNN network.

4.1.1 Pose Estimation

Based on the findings presented in [11], the estimated 2D keypoints consistently outperform both sensor-collected (NTU-60 [29]) and estimated 3D keypoints in action recognition task. Accordingly, in our approach, we leverage a top-down pose estimator to extract precise human 2D keypoints. We opt for a one-stage pose estimator to optimize time efficiency because of our objective of constructing an efficient pipeline for real-world applications. The pose estimator receives input frames and outputs bounding boxes and coordinate-triplets, *i.e.* (x, y, score), of each keypoint. We have a total of 17 coordinate-triplets for each person based on COCO-keypoints. We train our model with the augmented dataset.

4.1.2 Pseudo Joint Heatmap Generation

To build a lightweight action recognition network, we need to reform the extracted 2D keypoints into joint heatmaps. First, after we extract the human 2D keypoints, we sample 48 frames uniformly and discard the remaining frames. Therefore, we have 48 coordinate-triplets (x_k, y_k, c_k),

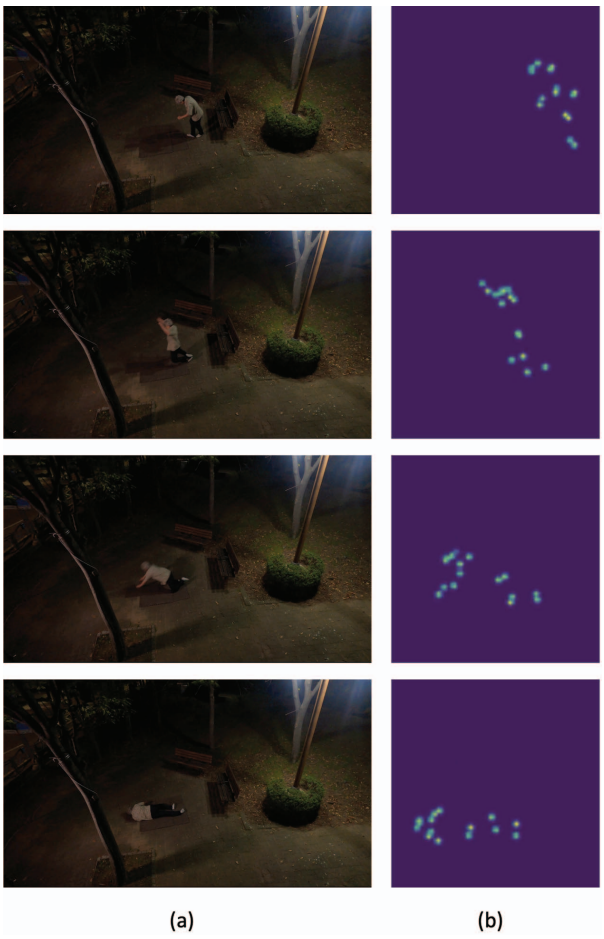

Figure 2: Examples of generated heatmaps through the frames. (a) is the RGB frame, while (b) is the generated pseudo joint heatmap.

which will be used as input to generate generate the joint heatmaps. As performed in [11], we generate Gaussian maps that are centered at every joint location,

$$\mathbf{J_{kij}} = c_k \cdot exp\{-\frac{(i-x_k)^2 + (j-y_k)^2}{2\sigma^2}\}, \qquad (1)$$

where (x_k, y_k) is the location, c_k is the confidence score of the k-th joint. σ controls the variance of Gaussian maps. Moreover, to make it more efficient, we perform subject-centered cropping. Given that the output of our one-stage pose estimator includes the bounding box location, we crop the frame based on the bounding box location and resize it to match the size of the input spatial setting. In our method, we use 56×56 as our spatial size. Examples of the generated joint heatmaps are shown in Figure 2.

Stage	Lightweight-AR	Output Sizes $T \times H \times W$
Data Layer	uniform 48, 56×56	$48 \times 56 \times 56$
Stem Layer	conv_1 $1 \times 7 \times 7$, 8	$48 \times 56 \times 56$
Res_2	$\begin{bmatrix} 1 \times 1 \times 1, 8 \\ 1 \times 3 \times 3, 8 \\ 1 \times 1 \times 1, 48 \end{bmatrix} \times 3$	$48 \times 56 \times 56$
Res_3	$\begin{bmatrix} 1 \times 1 \times 1, 16 \\ 1 \times 3 \times 4, 16 \\ 1 \times 1 \times 1, 64 \end{bmatrix} \times 4$	$48 \times 28 \times 28$
Res_4	$\begin{bmatrix} 3 \times 1 \times 1, 48 \\ 1 \times 3 \times 3, 48 \\ 1 \times 1 \times 1, 128 \end{bmatrix} \times 6$	$48 \times 14 \times 14$
Res_5	$\begin{bmatrix} 3 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$	$48 \times 7 \times 7$
	GAP, FC	# of Classes

Table 1: Lightweight action recognition network. The dimensions of the kernels are denoted by $T \times H \times W, C$ for temporal, spatial (height and width), and the channel size. GAP and FC denotes the global average pooling and fully connected, respectively.

4.1.3 Lightweight Action Recognition

The overall architecture of our model is shown in Figure 3. The input to the action recognition network is obtained by stacking the joint heatmaps along the temporal dimension with the size of $K \times T \times H \times W$ where K is the number of joints, T is the temporal dimension, and H, W are the height and width of the frame, respectively. The detailed network structure is listed in Table 1, which is a temporally strided 3D-CNN modified from the slow pathway of [13]. Considering that our goal is to integrate this model into our real-time system, we need the model to be as light as possible to accelerate the processing time while maintaining accuracy.

To this end, we use 48 input frames to accelerate the inference process. Subsequently, we reduce the width-channels from 64 to 8 and the spatial size from 224 to 56. This reduction is sufficient, given that our input consists of joint heatmap sequences, which do not require the same size as RGB frames. Our experiments show that this reduction can significantly reduce training time by up to 50% while still maintaining high performance, despite the lighter backbone.

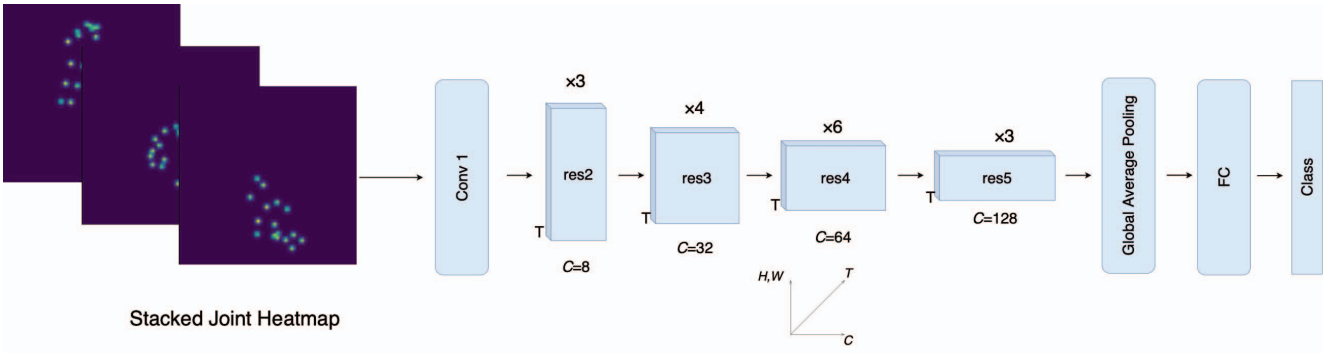

Figure 3: Architecture of the lightweight 3D-CNN action recognition. The input into our action recognition model is stacked joint heatmaps, which have the size $K \times T \times H \times W$, where K is the number of joints, T is the number of frames in our method we use 48 frames, H and W is the height and width of the frame, respectively.

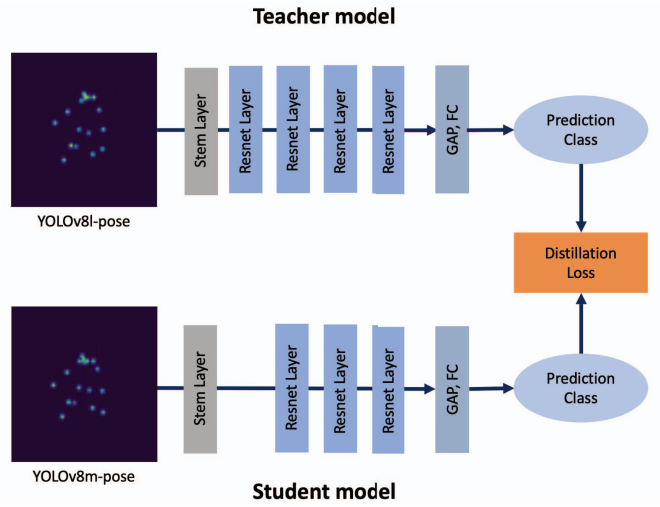

Figure 4: Knowledge distillation pipeline.

4.2. Real-Time Fall Detection and Action Recognition System

For each frame captured by the camera, our process starts with the pose estimation model, which extracts 2D keypoints and bounding boxes for each detected person. These output bounding boxes are then used as inputs for the tracking module [34], which ensures smooth and continuous tracking of each person across frames.

After the identities of individuals are established, we gather the keypoints for each person over a consecutive sequence of 48 frames, which results in a data structure of $K \times T$ (where K represents the number of keypoints, and T denotes the temporal length of 48 frames). This concatenated information is then fed into our action recognition model. With all the necessary data in place, our action recognition model performs action classification for each individual. We show examples of the results of our proposed inference frames in Figure 5.

Furthermore, in real-time inference for single-person scenario, we achieve an average FPS of 26 Hz. For multi-person scenarios, we achieved an average FPS of 13 Hz, which further showcase the efficiency and feasibility of our approach for real-world applications.

4.3. Knowledge Distillation

In order to accelerate the inference processing time, we employ the knowledge distillation framework shown in Figure 4. In this approach, the lightweight 3D-CNN model with 4 residual layers acts as the "Teacher," while a smaller network that has 3 residual layers and a different extracted pose serves as the "Student". We employ response-based knowledge [16] which focus on the output of the teacher model, such that the student will mimic its prediction. By leveraging this knowledge distillation technique, we aim to transfer the knowledge from the Teacher to the Student network. This way enables the latter to achieve comparable performance while being more computationally efficient.

5. Experimental Results

In this section, we present the experimental results of our proposed method. First, we validate whether our model can effectively distinguish falling motion from other daily activity motions. Second, we conduct a comprehensive comparison between our action recognition model and pipeline with the baseline method.

5.1. Implementation Details

In this work, we utilize YOLOv8-pose as our pose estimator. This is mainly due to its state-of-the-art real-time object detection capabilities. We first, train YOLOv8-pose with our proposed augmented human pose dataset. Then, we extract the 2D keypoints of each dataset with the trained

Figure 5: Example frames of the proposed multi-person fall detection and action recognition system.

YOLOv8-pose. We also train the baseline method [11] using the same dataset to ensure a fair comparison.

Our model and the baseline model [11] are trained for 250 epochs to ensure sufficient convergence and learning. We run our inference on a single RTX 4090 GPU.

5.2. Evaluation on Fall Detection

We evaluate our methods on two benchmark datasets for fall detection and compared them with previous fall detection methods. In addition, we perform another experiment using the AI Hub dataset to compare the performance of our network with the baseline method.

5.2.1 Datasets

UR Fall Detection (URFD) Dataset [18] contains 70 videos: 30 videos of falling motion from 2 different camera angles and 40 videos of activities of daily living. For training and testing, we use a random dataset split with an 80:20 ratio for each class.

Multiple Cameras Fall (Multicam) Dataset [5] contains 24 scenarios captured by 8 cameras, which results in 192 videos, including a total of 9 action classes. We separate action sequences in each video to generate a total of 1,304 videos. Considering that no official data split is given, we split the dataset by cross-view split: Camera #1 through #7 as the training set and camera #8 as the test set.

Model	Accuracy	Specificity	Sensitivity
Lin *et al.* [21]	92.00	-	-
Hasan *et al.* [17]	-	96.00	99.00
Marcos *et al.* [27]	95.00	92.00	100.00
Salimi *et al.* [28]	98.90	-	100.00
Ours	**100.00**	**100.00**	**100.00**

Table 2: Comparison of results on the URFD dataset [18].

AI Hub Senior Abnormal Behavior Dataset [1] contains a total of 9,400 videos of 3 human actions. The actor is assumed to be elderly and the action classes are threefold: fall down, wander, and dementia. We discard the dementia class, and only use the falling and wander data.

Model	Accuracy	Specificity	Sensitivity
Feng *et al*. [15]	-	93.50	91.60
Fan *et al*. [12]	-	97.90	97.10
Marcos *et al*. [27]	-	96.20	**98.07**
Hasan *et al*. [17]	97.38	96.00	98.00
Lu *et al*. [25]	99.07	99.56	86.21
Ours	**99.39**	**100.00**	96.15

Table 3: Comparison of results on the Multicam dataset [5].

Model	Accuracy	Specificity	Sensitivity
PoseConv3D [11]	97.56	97.72	**97.26**
Ours	**99.02**	**100.00**	**97.26**

Table 4: Comparison of results on the AI Hub dataset [1].

5.2.2 Evaluation Metrics

Fall detection is a binary classification task, and we adhere to the evaluation protocol from prior work [2]. *Accuracy* is the proportion of correctly detected falls and non-fall behaviors. *Sensitivity* is the proportion of correctly detected falls in all fall events. *Specificity* is the proportion of correctly detected non-fall behaviors in all the non-fall events. All metrics are represented in % in this paper.

5.2.3 Comparison with the State-of-the-Art

We begin to evaluate the performance of our fall detection by conducting experiments using a benchmark fall detection dataset. The outcomes are summarized in Table 2, where we compare our method on the URFD dataset [18] against other existing approaches. Impressively, our proposed system outperforms all other methods by achieving 100% for all evaluation metrics.

Furthermore, we evaluate our method on the Multicam dataset [5], and the results are presented in Table 3. Again, our proposed system exhibits superior performance to previous works. These comprehensive evaluations affirm the effectiveness of our proposed model in accurately distinguishing falling motion, which surpasses the performance of existing approaches by a significant margin. The prior works' results in Table 2 and Table 3 are obtained from their original papers.

Moreover, we compare our model with the baseline method [11] using the AI Hub dataset [1]. The results, as presented in Table 4, show that our pipeline has improved accuracy compared with the baseline method [11].

5.3. Evaluation on Action Recognition

To analyze the performance and provide a fair comparison with benchmark algorithms, we first follow the training and evaluation protocols used in the baseline method.

5.3.1 Datasets

5 action combined dataset is a compilation of various datasets merged together to be employed in our action recognition system. In total we have a total of 6,231 videos. We define 5 action classes: Falling, Sit Down, Stand Up, Walking, and Laydown. The details of the video sources of each class are shown in Table 5. We subsequently divide this dataset into training and test sets using an 80:20 ratio for each class to facilitate training.

5.3.2 Comparison with the Baseline

The proposed method is evaluated with the combined dataset and the baseline method [11]. Table 6 shows the quantitative results, which demonstrate that our augmented human pose dataset significantly improves the accuracy of the action recognition model. It highlights the dependence of skeleton-based approaches on the pose estimator. Despite having 4 times fewer trainable parameters and a processing time 10 times faster, the accuracy of our model remains on par with that of the baseline method trained with our pipeline.

Finally, we show the qualitative results[1] of our method on multiple datasets in Figure 5. These results show that our method performs well, able to detect falling motion and classify other action classes correctly.

5.3.3 Knowledge Distillation Result

For performance comparison, we first train the teacher model (4 residual layers trained with extracted keypoints from YOLOv8l-pose) and the student model (3 residual layers trained with extracted keypoints from YOLOv8m-pose) separately. Then, we train the student using the knowledge distillation technique. As shown in Table 7, the FPS of the student model is higher than that of the teacher model. Moreover, the standalone student model, which is not trained using KD techniques, exhibits lower accuracy.

5.4. Ablation Study on Pose Extraction

In skeleton-based approaches, choosing the right pose estimator is important; consequently, for a real-time system, a fast and precise pose estimator is needed. We compare the speed between the two-stage pose estimator HRNet [31] that is used in the baseline method [11] and one-stage YOLOv8-pose as shown in Table 8. Using an one-stage pose estimator has a faster processing time while also having a better accuracy.

[1] Video results available at https://youtu.be/6zWSdxvnigg

Dataset	Falling	Sit down	Stand up	Walking	Lay down	Total
NTU RGB+D [29]	948	948	948	-	-	2,844
NW-UCLA [33]	-	148	149	173	-	470
URFD [18]	60	-	-	-	-	60
Multicam dataset [5]	200	208	232	264	216	1,120
AI Hub [1]	559	-	-	1,178	-	1,737
Total	1,767	1,304	1,329	1,615	216	6,231

Table 5: Source details of the 5 action combined dataset.

Action Recognition Model	Pose Estimator Model	Human Pose Dataset	Accuracy (%)	Parameters (M)	Action Proc. Time (milliseconds)
PoseConv3D [11]	HRNet [31]	COCO-pose only [22]	88.06	2.0	392.50
PoseConv3D [11]	YOLOv8-pose	Augmented Pose Dataset	98.26	2.0	392.50
Ours	YOLOv8-pose	Augmented Pose Dataset	**97.93**	**0.5**	**40.00**

Table 6: Comparison of results on the 5 action combined dataset. The first row represents the baseline method with its own pipeline. It involves extracting the dataset using Faster-RCNN [30] as the detector and HRNet [31] as the pose estimator. Both are trained on the COCO-person dataset [22]. In the second row, the baseline method is trained with our pipeline, which utilizes YOLOv8-pose trained on our augmented human pose dataset.

Model	Accuracy	Parameters (M)	Avg. FPS
Teacher	**97.93**	0.5	26.40
Student	87.67	**0.1**	29.23
Knowledge Distillation	89.59	**0.1**	**31.40**

Table 7: Comparison of knowledge distillation on the 5 action combined dataset.

Model	Person Detection	Pose Estimator	Total
Faster-RCNN + HRNet	66.7	87.0	153.7
YOLOv8-pose	-	**20.0**	**20.0**

Table 8: Pose estimator speed comparison (in microseconds).

6. Limitations

There are a couple of limitations of this work. First, the speed of is directly affected by the number of persons present in the frames, which leads to slower processing time with an increasing number of individuals. Second, although our network performs well for small classes, it may be unsuitable for systems with a higher number of classes.

7. Conclusion

In this paper, we developed a solution to the limitations of existing skeleton-based action recognition for detecting falling motion in real-time. In addition to the newly augmented human pose dataset, we proposed a skeleton-based 3D-CNN lightweight network with less parameters. On top of those, we developed an efficient real-time multi-person fall detection and activity recognition system. The result showed significantly improved performance in terms of accuracy and speed.

Future work includes implementing the proposed system on the actual embedded platform for practical application. Due to the difference between the development environment and GPU on desktop computers and embedded platforms, a lot of engineering tasks are expected to be completed to make this happen.

Acknowledgement

We thank Dr. Hyun Su Hong in Difine for the constructive discussions and support during the project.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University) and No.2022-0-00981, Foreground and Background Matching 3D Object Streaming Technology Development and No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] AI Hub Senior Abnormal Behavior Video Dataset. https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=167, 2020. 3, 6, 7, 8

[2] Ekram Alam, Abu Sufian, Paramartha Dutta, and Marco Leo. Vision-based human fall detection systems using deep learning: A review. *CoRR*, abs/2207.10952, 2022. 1, 7

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2014. 3

[4] Andrea Apicella and Lauro Snidaro. Deep neural networks for real-time remote fall detection. In *Pattern Recognition. ICPR International Workshops and Challenges*, volume 12662 of *Lecture Notes in Computer Science*, pages 188–201. Springer, 2020. 3

[5] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Multiple cameras fall data set. *Technical report 1350, DIRO - Université de Montréal*, 2010. 6, 7, 8

[6] Djamila Romaissa Beddiar, Mourad Oussalah, Brahim Nini, and Yazid Bounab. Vision-based fall detection using body geometry. In *Pattern Recognition. ICPR International Workshops and Challenges*, volume 12664 of *Lecture Notes in Computer Science*, pages 170–185. Springer, 2020. 3

[7] Yong Chen, Weitong Li, Lu Wang, Jiajia Hu, and Mingbin Ye. Vision-based fall event detection in complex background using attention guided bi-directional LSTM. *IEEE Access*, 8:161337–161348, 2020. 2

[8] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 180–189, 2020. 2

[9] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-CNN: Pose-based CNN features for action recognition. In *Proc. IEEE International Conference on Computer Vision*, pages 3218–3226, 2015. 2

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015. 2

[11] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2959–2968, 2022. 2, 3, 4, 6, 7, 8

[12] Yaxiang Fan, Martin D. Levine, Gongjian Wen, and Shaohua Qiu. A deep neural network for real-time detection of falling humans in naturally occurring scenes. *Neurocomputing*, 260:43–58, 2017. 3, 7

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proc. IEEE/CVF International Conference on Computer Vision*, pages 6201–6210, 2019. 2, 4

[14] Pengming Feng, Miao Yu, Syed Mohsen Naqvi, and Jonathon A. Chambers. Deep learning for posture analysis in fall detection. In *Proc. International Conference on Digital Signal Processing*, pages 12–17, 2014. 2

[15] Qi Feng, Chenqiang Gao, Lan Wang, Yue Zhao, Tiecheng Song, and Qiang Li. Spatio-temporal fall event detection in complex scenes using attention guided LSTM. *Pattern Recognition Letters*, 130:242–249, 2020. 2, 7

[16] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 5

[17] Md Mahedi Hasan, Md Shamimul Islam, and Sohaib Abdullah. Robust pose-based human fall detection using recurrent neural network. In *Proc. IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things*, pages 48–51, 2019. 3, 6, 7

[18] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014. 6, 7, 8

[19] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Haoshu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10863–10872, 2019. 3

[20] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019. 2

[21] Chuan-Bi Lin, Ziqian Dong, Wei-Kai Kuan, and Yung-Fa Huang. A framework for fall detection based on openpose skeleton and LSTM/GRU models. *Applied Sciences*, 11(1), 2021. 3, 6

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 3, 8

[23] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C. Kot. Global context-aware attention LSTM networks for 3D action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3671–3680, 2017. 2

[24] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 140–149, 2020. 2

[25] Na Lu, Xiaodong Ren, Jinbo Song, and Yidan Wu. Visual guided deep learning scheme for fall detection. In *Proc. IEEE Conference on Automation Science and Engineering*, pages 801–806, 2017. 3, 7

[26] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2D/3D pose estimation and action recognition using multitask deep learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018. 2

[27] Adrián Núñez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. Vision-based fall detection with con-

volutional neural networks. *Wireless Communications and Mobile Computing*, 2017, 2017. 3, 6, 7

[28] Mohammadamin Salimi, José J. M. Machado, and João Manuel R. S. Tavares. Using deep neural networks for human fall detection based on pose estimation. *Sensors*, 22(12):4544, 2022. 3, 6

[29] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016. 3, 8

[30] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *Proc. ACM International Conference on Multimedia*, pages 1625–1633, 2020. 8

[31] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 7, 8

[32] Tzutalin. LabelImg. `https://github.com/tzutalin/labelImg`, 2015. 3

[33] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 8

[34] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proc. IEEE International Conference on Image Processing*, pages 3645–3649, 2017. 5

[35] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proc. AAAI Conference on Artificial Intelligence*, pages 7444–7452, 2018. 2