Multi-level Recognition on Falls from Activities of Daily Living

Jiawei Li[*]
Department of Computer Science and Technology, Tsinghua University
Beijing, China
li-jw15@mails.tsinghua.edu.cn

Shu-Tao Xia
Center of Communications and Networks, PengCheng Laboratory
Shenzhen, China
xiast@sz.tsinghua.edu.cn

Qianggang Ding
Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
dqg18@mails.tsinghua.edu.cn

ABSTRACT

The falling accident is one of the largest threats to human health, which leads to broken bones, head injury, or even death. Therefore, automatic human fall recognition is vital for the Activities of Daily Living (ADL). In this paper, we try to define multi-level computer vision tasks for the visually observed fall recognition problem and study the methods and pipeline. We make frame-level labels for the fall action on several ADL datasets to test the methods and support the analysis. While current deep-learning fall recognition methods usually work on the sequence-level input, we propose a novel Dynamic Pose Motion (DPM) representation to go a step further, which can be captured by a flexible motion extraction module. Besides, a sequence-level fall recognition pipeline is proposed, which has an explicit two-branch structure for the appearance and motion feature, and has canonical LSTM to make temporal modeling and fall prediction. Finally, while current research only makes a binary classification on the fall and ADL, we further study how to detect the start time and the end time of a fall action in a video-level task. We conduct analysis experiments and ablation studies on both the simulated and real-life fall datasets. The relabelled datasets and extensive experiments form a new baseline on the recognition of falls and ADL.

CCS CONCEPTS

• **Computing methodologies → Activity recognition and understanding**; *Object detection*; • **Applied computing → Health care information systems**.

KEYWORDS

fall detection, activities of daily living, pose motion representation

ACM Reference Format:
Jiawei Li, Shu-Tao Xia, and Qianggang Ding. 2020. Multi-level Recognition on Falls from Activities of Daily Living. In *Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR '20), June 8–11, 2020, Dublin, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3372278.3390702

1 INTRODUCTION

With the aging of the population, the health and life problem of the elders becomes more and more critical. According to the statistics [28], the accidental fall is believed to be the leading cause of serious injuries for the elderly. Moreover, fall is the leading cause of accidental death in seniors aged 65 and above [4]. Therefore, there is a great need to develop the methods to detect the fall from activities of daily living (ADL) and timely raise the alarm. To date, there exists a lot of fall detection methods [18]. Many of them are relying on additional sensors [16], such as multiple RGB cameras [10], depth sensors [15], accelerometers or smart watches [16]. Compared to the wearable fall detection methods, the vision-based method has a significant advantage because it does not require the seniors to wear specific equipment. The widespread existence of indoor surveillance systems also facilitates the application of vision-based human fall recognition. This paper try to explore the feasible vision-based fall detection methods, which works on the recorded videos or the real-time surveillance video stream.

To detect a fall from the RGB camera, many previous methods [18] try to leverage the shape deformation and motion feature of the human body. However, ADL might have a similar body movement to the motion feature of fall action [4]. For example, sitting on knees, bending forwards, laying down and sleep in the low-level visual feature looks like the accidental fall, but current shape deformation based method [20] is unable to distinguish the similar motion patterns. Recently, a deep learning based fall detection method [13] utilizes the dynamic image [5] and a temporal modeling pipeline to make the fall detection on the sequence-level input. However, the dynamic image only focuses on the motion information and disregards the pose appearance, which is a limited feature representation. Besides, the temporal modeling procedure in [13] is highly dependent on an elaborate decomposition of the fall action. It defines a fall action with four smaller atomic action units (such as standing, falling, fallen, and not-move) and requires the accordingly frame-level labels. As the boundary between adjacent atomic units (e.g., falling and fallen) is quite vague, making temporal modeling with them is also limited.

To address these limitations, in this paper, we propose to make the fall recognition on multiple levels, as shown in Figure 1. We make these multi-level task descriptions similar to canonical multi-level human action recognition methods, such as the short-term [25], mid-term [27], and long-term [26] modeling methods. The short-term modeling usually takes around 15 frames as input, while the mid-term takes around 10 seconds as input. However, due to the fall action is quite fast than the other ADL, our multi-level tasks process more fine-grained inputs than the other human action recognition methods.

[*] The author's work was made when visiting Tencent AI Lab.

ICMR '20, June 8–11, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-7087-5/20/06…$15.00
https://doi.org/10.1145/3372278.3390702

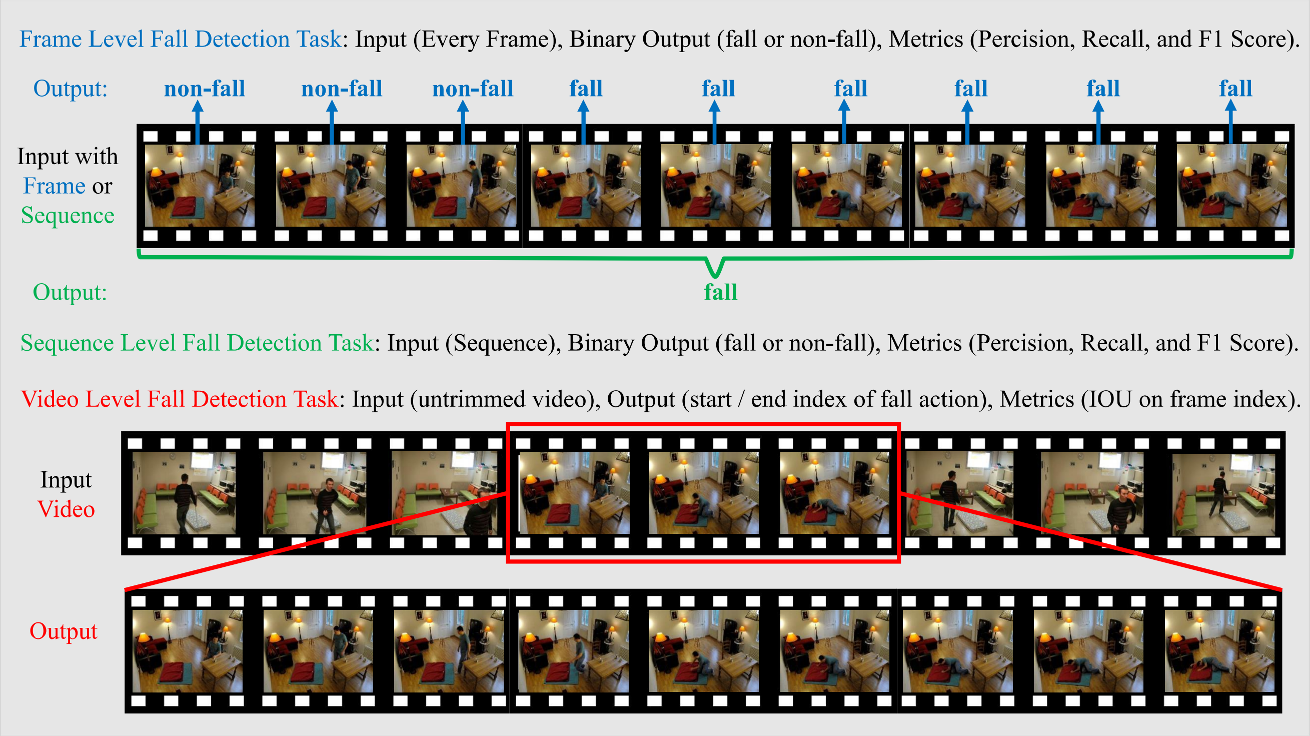
Frame Level Fall Detection Task: Input (Every Frame), Binary Output (fall or non-fall), Metrics (Percision, Recall, and F1 Score).
Output:
non-fall
non-fall
non-fall
fall
fall
fall
fall
fall
fall
Input with Frame or Sequence
Output:
fall
Sequence Level Fall Detection Task: Input (Sequence), Binary Output (fall or non-fall), Metrics (Percision, Recall, and F1 Score).
Video Level Fall Detection Task: Input (untrimmed video), Output (start / end index of fall action), Metrics (IOU on frame index).
Input Video
Output

Figure 1: We define multi-level computer vision tasks on the visually observed fall action. To better evaluate the final detection results, we calculate the precision, recall, F1 score, and IOU value to form the performance baseline.

After clarifying the different objectives of the multi-level fall detection tasks, we also benefit from addressing them. If the threshold value of the fall judgment is low, we will have a high recall rate on the fall frames. Therefore, the methods for Frame Level Fall Detection (FLFD) are suitable for mining the potential fall action. When the input is in sequence-level, we try to find a more effective fall detection algorithm pipeline to be comparable with previous practices. Because an untrimmed video is similar to a real-time stream, to some extent, an application of Video Level Fall Detection (VLFD) is to explore how to apply the fall detection method into a surveillance system.

As a most powerful deep learning method, the convolutional neural network (CNN) has been widely used in the field of human action recognition (HAR) and fall recognition problem [30]. At present, a typical video-processing algorithm pipeline uses CNN to extract the visual features [19] and adopts the spatial-temporal two-stream [21], 3D convolution [24], or the CNN-LSTM [12] model architectures. Technically, in addition to proposing a new fall detection pipeline, finding a better motion feature [13] or pose feature [14] is very important for the fall detection. A recent study [8] shows that the combination of 2D pose heatmaps and its motion is a more informative action feature. Based on this observation, in this paper, we propose a more powerful feature to characterize the fall action, which is the dynamic pose motion (DPM). We will introduce the theory, advantages, and the usage of DPM in the Sequence Level Fall Detection (SLFD) task.

Although previous research develop a lot of vision-based human fall detection algorithms [13] [14] [22], there is a gap between the real-life and the simulated human fall action [4] in the visual presentation. To better evaluate the proposed fall detection methods for different scenarios, we collect five public available fall datasets and divide them into two categories: the simulated and the real-life fall videos. There are 483 laboratory simulated videos for the standard evaluation and 379 real-life accidental fall videos for the real application test. We defined different evaluation metrics for multi-level tasks. The algorithm described in [13] is regarded as the compared benchmark. To show the performance improvement of pose and motion information, we also conduct the ablation studies.

The main contributions of this paper can be summarized as follows:

- We make a multi-level task definition for the fall recognition problem. In this way, we can focus on different difficulties and address the problem from different perspectives.
- For the widely concerned SLFD task, we propose a novel dynamic pose motion (DPM) feature to represent the human fall action. From the ablation study, we show the DPM is the most suitable spatial and temporal feature representation for the fall action at present.
- We carefully relabel five representative fall detection datasets in frame-level and provide baselines on the simulated or real-life datasets. The ablation study on different human pose and motion features are also conducted.

2 RELATED WORK

2.1 Action Modeling Methods in ADL

There exists many action recognition/classification methods in the ADL scenario. The short-term modeling method ARTNet [25] extracts the appearance and motion from every video frames. It has a canonical two-stream architecture, and requires to extract the optical flow in advance for every frame, which means the motion branch limited the detecting speed. If the motion branch can speed up, the ARTNet method could be powerful for frame-level fall detection. The temporal segment network (TSN) [27] processes the middle-range feature of action, and it is a sequence-level recognition framework based on a simple segment sampling strategies and consensus aggregation. In general, TSN is a powerful and flexible framework for action modeling. However, its temporal convolution branch takes the stacking warped optical field as the input, which is insufficient for the motion information of human fall [13]. The UntrimmedNet [26] makes the weakly supervised action recognition on the video-level. It is an end-to-end architecture and combines the feature extraction, sequence selection, and classification modules. Due to the advantage of end-to-end and video-label prediction, the whole pipeline is convenient for action recognition or detection.

There exist the short-term, middle-term, and long-term modeling models for the action recognition problem, but there is no time-aware distinction from the task definition of the fall recognition from ADL. Recent research [4] provides a High Quality Fall Simulation (HQFS) dataset. It has a long recording session on the performance of the data acquisition system, instead of recording only the short segments. By this way, the dataset envisions a better balance between the fall action and the ADL and can be a more reliable benchmark. In this paper, we do not provide a new dataset, but we propose the multi-level definitions for fall detection. Accordingly, we also offer the relevant new labels for several widely used fall datasets in frame-level, sequence-level, and video-level.

2.2 Vision-based Fall Recognition

A traditional recognition system requires additional wearable sensors to detect a human fall action [16]. However, recent years have seen rapid progress in computer vision. Thus there are more and more researches concerning the vision-based fall detection algorithms [13]. The vision-based fall detection system usually utilizes a surveillance camera to capture a video stream and makes the SLFD task.

A typical technique [9] detects the person in the video sequence firstly, and then compute three points that represent different regions of a human body, namely the head, body, and legs from the detected foreground. This method is simple, but it can only recognize one person, and the performance suffered in the presence of other objects, such as walking sticks.

Since 3D camera provides depth information, the depth camera (e.g., the Microsoft Kinect sensor) is also used for video capturing and fall detection. [23] uses the Kinect for falling detection. Although the 3D cameras can make a more robust detection, the visual range of depth camera is usually shorter than the 2D camera. For example, Kinect can only detect human motion from 0.4 meters to 3 meters, while most of the 2D RGB cameras detect farther than 3 meters. [3] provides a multi-view fall dataset for developing the multi-camera fall detection algorithm. The multi-camera fall detection system can exploit the enriched visual information for action description, but it has a high cost, and it is limited for the indoor scenario. In this paper, we propose a single 2D RGB camera-based fall recognition algorithm, and it is expansible to the other existing fall detection systems.

2.3 Pose Estimator

Intuitively, the human pose is a very discriminate cue for human action recognition. Recent trend and several approaches [6][1][29] propose to use a CNN to extract the human pose from 2D RGB video. The OpenPose [6] is the first real-time multi-person system to detect the joint (18 keypoints from OpenPose COCO) of the human body from a video. The core algorithm of OpenPose has a two-stage greedy bottom-up parsing. At first, it can efficiently find the heatmap of joint keypoints, then the part affinity fields (PAFs) help to connect body parts with individuals. To make the fall detection, we only need the first stage of OpenPose to detect the human body. The PoseFlow [29] has a multi-stage pipeline by separating the pose detection task into body detection, single person pose estimation, and post-processing operations. Using better human detector and single person pose estimator network, the PoseFlow has a state-of-the-art performance in several benchmark pose datasets. The DensePose [1] maps all human pixels of 2D RGB images to a 3D surface-based model of the body. Because the DensePose almost provides a very accurate rendering for the human surface, it refers to a new task called the dense human pose estimation. It has a fully-convolutional dense pose regression and determines which surface part the pixels belonging. By providing a pixel-level mask for the human pose feature, it is quite robust for the real-life fall detection scenario.

The pose representation provides a most directly feature for the fall recognition from ADL [11][14], but it still has several limitations: (1) the time-consumption of pose tracking across the video is quite high, (2) the joint feature maps are not always correct, (3) current pose estimators are not robust enough to the occlusion and truncation. To avoid these defections, we propose a sequence processing module to utilize the motion pose information [8]. We implement it for the SLFD task and achieve the state-of-the-art performance.

2.4 Pose Motion Representation

To date, the spatiotemporal action recognition usually has two branches (or two-streams) [21]. A spatial branch is typically used to extract the appearance information from the RGB frames, while the temporal branch is applied to characterize the motion information using the optical flow. In many human action recognition tasks, the interested area is the human body but not the global. Optical flow for the whole video will bring in some noisy information, such as the irrelevant background object movement. To avoid this problem, another approach [13] simultaneously describes both of the appearance and temporal information in a serial pipeline, which converts a short video clip to a dynamic image [5]. Different from the optical flow, the dynamic image employs a rank learning method to combine the frames of a video into a single image to enable the action analysis.

In addition to extract the dynamic image [5] feature for human action, there exists a PoTion [8] representation, which relies on the human pose and is also complementary to the two-stream approach. Furthermore, both of the dynamic image and PoTion can capture long-term dependencies without any limitation on the temporal receptive field. In practice, we approximate the dynamic image by weighting frames with the time index. We obtain the PoTion by aggregating all frames' pose heatmaps into a colorized heatmap. Due to all the calculation are the simply weighting, these two motion representations are easy to implement with particular pooling layers. The low time-consumption of pooling makes the dynamic image scalable for quick fall action [5]. In this paper, we extend the PoTion representation and propose the novel Dynamic Pose Motion (DPM) feature for multi-level fall recognition.

3 MULTI-LEVEL FALL RECOGNITION

In this section, we first describe the multi-level fall recognition tasks. Then we introduce the novel Dynamic Pose Motion (DPM) feature and design three algorithm pipelines for FLFD, SLFD, and VLFD tasks, respectively. Notably, as we have labeled the datasets on the frame-level, the VLFD task can benefit from the backbone algorithm of FLFD and SLFD.

3.1 Overview for Multi-level Tasks

According to the different task purpose, technically, we define the fall recognition problem from the frame-level, sequence-level and video-level. The detailed differences are listed on the Table 1.

The **frame-level fall detection (FLFD)** task takes the single frame as the input and judges whether there exists a person falls in this frame. To solve this problem, we need to preprocess the dataset and give every frame a *fall* or *non-fall* label. The adopted detection method should judge the fall action within the single frame and do not depend on other frames. How to extract a better body appearance feature from the still image is essential for the FLFD. To improve the performance of a FLFD method, we can focus on finding better pose representations.

The **sequence-level fall detection (SLFD)** task takes a fixed sequence of frames (e.g., 60[1] frames) as the input and judges whether this sequence contains a complete fall action. Recently, a typical approach [13] try to extract the dynamic image [5] as a motion feature of the input; and classify the feature from four atomic action units: standing, falling, fallen, no-move. After that, it makes a sequence learning to make the final decision on fall or non-fall. To better do the SLFD, we need to extract a better motion feature and design a better algorithm pipeline. In this paper, we propose a novel DPM feature and show the effectiveness of the CNN-LSTM [12] pipeline.

The **video-level fall detection (VLFD)** task takes the untrimmed video as the input and judges whether the people in this video has a fall action. Different from the SLFD, the VLFD requires a temporal processing strategy to localize the fall moments [2]. Therefore, we need to pre-process the input video and find all possible fall snippets for doing SLFD. In this paper, we utilize *sliding window* method and *shot-based sampling* method for the temporal processing. As an untrimmed video is very similar to a real-time stream, it is potential to transfer the VLFD method for a surveillance system.

Table 1: Multi-level fall recognition tasks.

	Task Objective	Feature	Method
FLFD	extract a better still appearance feature	body shape, pose, etc.	binary classifier
SLFD	design better motion feature and pipeline	DI, PoTion, DPM, etc.	cnn-lstm, two-stream
VLFD	temporal processing finds the fall snippet	DI, PoTion, DPM, etc.	long-term modeling

With the multi-level task definitions, we can avoid some ambiguous cases. For example, in the FLFD task, it is quite hard to distinguish *lay down*, *sleep*, and fall from the still frame. But we can better solve it, under the SLFD or VLFD task. The *fake fall* and *stagger* are the hard cases in the SLFD task, but we can also better discriminate them with the VLFD setting. Although most of the simulated fall cases could be solved in the VLFD task, making fall recognition in the wild is also a hard problem. In this paper, we establish the compared baselines with 379 real-life videos with a typical method [13] and our proposed method.

3.2 The Dynamic Pose Motion Representation

We will introduce a novel pose motion representation here, it is named as Dynamic Pose Motion (DPM) and is closely related to the dynamic image [5] and the PoTion [8] representation.

3.2.1 *Dynamic Image.*

The dynamic image is an efficient method to present the motion features of a sequence of frames to a still image. The basic idea of dynamic image is ranking the feature (such as the human pose heatmap) of frames ($\psi_1, ..., \psi_T$) of a T-length video. Define the time average of ψ_t is $V_t = \frac{1}{t}\sum_{\tau=1}^{t}\psi_t$. Then a ranking score associated with time t is represented by $S(t|\mathbf{d}) = \langle \mathbf{d}, V_t \rangle$. Due to the temporal dynamic information of video, we restrict the ranking score to a positive value, where $\forall\{q, t\}\ s.t.\ q > t \Rightarrow S(q|\mathbf{d}) > S(t|\mathbf{d})$. Then we can solve $\mathbf{d}$ from the following convex optimization problem:

$$\mathbf{d}^* = \rho(\psi_1, ..., \psi_T) = \arg\min_{\mathbf{d}}(L(\mathbf{d}) + \frac{\lambda}{2}||\mathbf{d}||^2) \tag{1}$$

$$L(\mathbf{d}) = \frac{2}{T(T-1)}\sum_{q>t}\max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\}. \tag{2}$$

This convex optimization problem converts a T-length video frames to a optimized frame-sized vector $\mathbf{d}^*$. This optimization process in [5] is named with *rank pooling*. When the optimized frame $\mathbf{d}^*$ is visualized as a standard RGB image, it will be called the *dynamic image*.

3.2.2 *Approximate Dynamic Image.*

To meet the requirement of real-time action recognition, a fast *approximate dynamic image* algorithm is introduced to speed up the optimization of dynamic images [5]. Considering the first step of gradient update of the optimization problem formula (1), we will have $\mathbf{d} = \vec{0}$. So the first-step approximated solution of the dynamic image will be $\mathbf{d}^* =$

[1] 60 is an empirical statistic value and calculated from 5 public available datasets.

$\vec{0} - \eta\nabla L(\mathbf{d})|_{\mathbf{d}=\vec{0}} \propto -\nabla L(\mathbf{d})|_{\mathbf{d}=\vec{0}}$ for $\eta > 0$, and

$$\nabla L(\vec{0}) \propto \sum_{q>t} \nabla\langle \mathbf{d}, V_t - V_q\rangle = \sum_{q>t} V_t - V_q. \tag{3}$$

$$\mathbf{d}^* \propto \sum_{q>t}(V_q - V_t) = \sum_{t=1}^{T} \beta_t V_t \tag{4}$$

The weighted parameter β_t in equation (3) is in terms of time t, and $\beta_t = (t-1) - (T-t) = 2t - T - 1$. Continue to expand the equation (3), then we will have:

$$\begin{aligned} \mathbf{d}^* &\propto \sum_{t=1}^{T} \beta_t V_t = \sum_{t=1}^{T} \alpha_t \psi_t \\ \alpha_t &= 2(T - t + 1) - (T+1)(H_T - H_{t-1}) \end{aligned} \tag{5}$$

where $H_t = \sum_{i=1}^{t} \frac{1}{t}$ is the t-th Harmonic number.

If we regard the α_t as a weight coefficient, the approximated dynamic image $\mathbf{d}^*$ will be calculated from weighting the original frames by time axis. In this way, the time-cost will be greatly reduced. Due to the approximated calculation can be implemented in a pooling layer, the dynamic image in the following text refer to the approximated dynamic image.

3.2.3 PoTion Representation. The PoTion [8] representation relies on the human pose and is complementary to the two-stream pipeline. At first, it colors the heatmap feature of a joint to obtain a colorized heatmap. Then the algorithm aggregates the different joints to generate a heatmap representation for the pose.

Specifically, after the extracting of joint heatmaps from each frame with OpenPose [6], they will be colorized to a time-dependent heatmap. The colorized heatmap of joint j for a pixel (x, y) and a channel c at time t is given by: $C_j^t[x, y, c] = \mathcal{H}_j^t[x, y]\mathbf{o}_c(t)$, where $\mathbf{o}_c(t)$ is a previously designated weight coefficient for the c-th color channel. The heatmap H_j^t of different positions of dimension $H\times W$ is transformed into an image C_j^t of dimension $H\times W\times C$ with the same spatial resolution but C color channels.

The aggregation of the colorized heatmaps for each joint j is $\mathcal{S}_j = \sum_{t=1}^{T} C_j^t$, which depends on the number of frames T. To obtain an invariant representation, we can normalize each channel c independently by dividing T or $\sum_t \mathbf{o}(t)$ over all pixels. Then the PoTion representation will be a C-channel image:

$$\mathcal{U}_j[x, y, c] = \frac{\mathcal{S}_j[x, y, c]}{\max_{x',y'} \mathcal{S}_j[x, y, c]}. \tag{6}$$

In addition to the PoTion $\mathcal{U}$, there are two more representations $\mathcal{I}$ and $\mathcal{N}$. The intensity image $\mathcal{I}_j$ is an image with a single channel:

$$\mathcal{I}_j[x, y] = \sum_{c=1}^{C} \mathcal{U}_j[x, y, c]. \tag{7}$$

Then a normalized PoTion representation is:

$$\mathcal{N}_j[x, y, c] = \frac{\mathcal{U}_j[x, y, c]}{\epsilon + \mathcal{I}_j[x, y]}. \tag{8}$$

where $\epsilon = 1$ can avoid instabilities in areas with low intensity.

While the pose heatmaps have been directly used as the fall feature [14], we can further use the PoTion representation to be the motion feature.

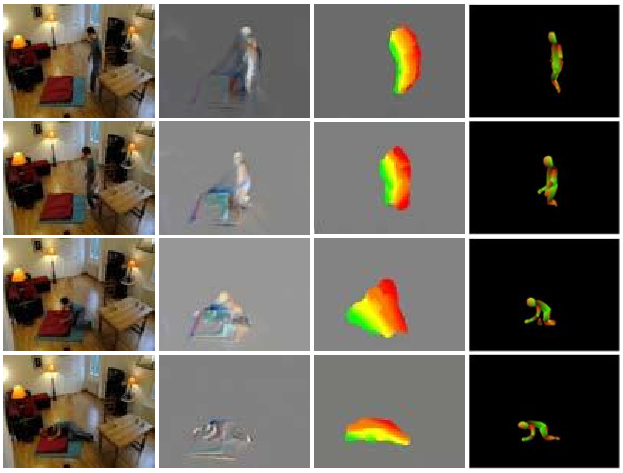

Figure 2: Compare the different pose motion representations on a sequence of fall frames. From left to right: the RGB frames, dynamic image, the proposed dynamic pose motion (DPM), and the normalized partial feature (NPF).

3.2.4 The Proposed Dynamic Pose Motion. Since the *PoTion* temporally aggregates the probability heatmaps with a weight coefficient and the *dynamic image* is a weighting operation on the feature maps. We propose a better pose motion representation by calculating the dynamic image and the PoTion representation simultaneously. We call this novel representation as **Dynamic Pose Motion (DPM)**.

Current 2D pose estimator, such as Openpose[6], outputs a heatmap to indicate the probability of each joint at pixel-level. Then the colorized pose heatmap $(C_j^1, ..., C_j^T)$ of a T-length video can be used to calculate $\mathcal{D}_j$, which is the DPM of the j-th pose joint. We can define a rank pooling optimization problem for the proposed dynamic pose motion:

$$\mathcal{D}_j^* = \rho(C_j^1, ..., C_j^T) = \arg\min_{\mathcal{D}_j}(L(\mathcal{D}_j) + \frac{\lambda}{2}||\mathcal{D}_j||^2) \tag{9}$$

where the ranking loss L keeps the same to equation (2).

Then we will still have a first-step gradient approximation $\mathcal{D}_j^* \propto \sum_{q>t}(V_j^q - V_j^t) = \sum_{t=1}^{T} \beta_t V_j^t$, where the weight $\beta_t = 2t - T - 1$ and the stacking $V_j^t = \frac{1}{t}\sum_{\tau=1}^{t} C_j^t$.

Compared to the *Dynamic Image*, the proposed DPM expend the obtained pose heatmap ψ into a colorized heatmaps C. The specific colorization schemes depend on the color channels number C and the definitions of $\mathbf{o}(t)$. For instance, the $\mathbf{o}(t)$ is $(\frac{t-1}{T-1}, 1 - \frac{t-1}{T-1})$ with $C = 2$. In this paper, we use $\mathbf{o}(t)$ with $C = 3$. Compared to the *PoTion* representation, the DPM further weighting on the temporal information for each joint. We can find this difference from the formula $\mathcal{S}_j = \sum_{t=1}^{T} C_j^t$ of PoTion and the approximated DPM formula $\mathcal{D}_j = \sum_{t=1}^{T} \beta_t V_j^t$.

We can refer to figure 2 for a visual comparison for different pose motion features. The normalized partial feature (NPF) is a combination of the *pose heatmap* and *DPM* feature. We first make the dot product for these two features, on the pixel level. Then we further normalize the obtained partial feature (similar to equation

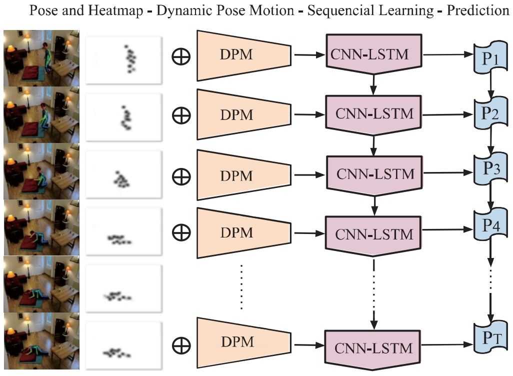
Pose and Heatmap - Dynamic Pose Motion - Sequencial Learning - Prediction
DPM
CNN-LSTM
P1
P2
P3
P4
PT

Figure 3: The sequence-level algorithm pipeline (SLAP) for SLFD task. Technically, we first extract the Pose feature and the DPM feature from a 2D RGB sequence. Then we train a CNN-LSTM on the stacked feature for the prediction.

8) on each body part. For a better fall recognition performance, we recommend using the NPF feature, as illustrated in figure 4.

3.3 Algorithm Pipelines for Multi-level Tasks

In this section, we describe the algorithm pipelines for multi-level tasks. They are the frame-level algorithm pipeline (FLAP), the sequence-level algorithm pipeline (SLAP), and the video-level algorithm pipeline (VLAP), respectively.

For the FLFD task, if the input frames are in a fall action, we desire that the algorithm pipeline output a big probability value p (bigger than a pre-assigned threshold) and say it fall. In this paper, we use the pre-trained pose estimators (e.g., Openpose [6], [29]) to extract the appearance feature, then use a simple binary classifier, such as a 4-layer MLP, to detect the fall frame.

For the SLFD problem, we first extract the pose and heatmap feature for every frame of the input sequence. Then, we calculate the DPM (default with $T = 15$) representation from the RGB frames and the pose features. We propose a typical CNN-LSTM structure to process the stacked feature, which is a concatation of the RGB frame, the Pose Heatmap, and the DPM feature. The final prediction result is a probability value p, and it is updated over the time index. The pipeline is also illustrated in figure 3.

For the VLFD task, we take an untrimmed video as input, and it might include indoor or outdoor scenes. Therefore we need to make the fall detection from a lot of the other activities daily living (ADL). When the datasets do no offer the frame-level fall labels, we can only adopt a *sliding window* method for the snippets sampling. We can not fine-tune the prediction model for the fall sequence recognition. Fortunately, as figure 4 shows, in our approach, we can sample human action snippets with the proposed FLFD models and make the *shot-based sampling*. To find the start index and the end index of a fall action, we make the sequential learning with the proposed SLFD models and aggregate [17] all the outputs for the adjacent snippets. In detail, we count the fall frame number n on continuous k frames. Only all the adjacent frames are fall frames (e.g., $n = k$), the current frame could be output as a fall frame.

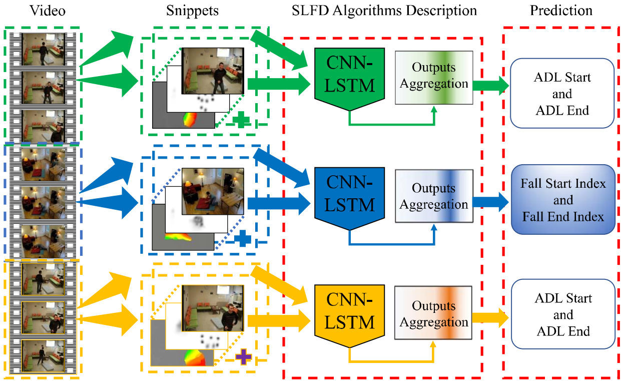

Figure 4: The video-level algorithm pipeline (VLAP) for VLFD task. Technically, for every action, we sample the interested snippets from the untrimmed input video. Then we extract the stacked pose and motion feature (or NPF) for every snippet. After that, we use the CNN-LSTM to make an ensemble for all the sampled snippets. Finally, if there is a fall action, we output the start index and the end index.

4 EXPERIMENTS

In this section, we show the influence of using different pose motion features and present the experimental results of the proposed algorithm pipelines on multi-level fall detection tasks.

4.1 Datasets

We establish the baselines on five public available fall detection datasets. Table 2 shows the detailed information of these datasets.

Multi-camera Fall Detection (MCFD) [3] is a simulated dataset. It contains 24 scenarios, and it is recorded with eight video cameras from a different perspective. There are 88 daily activities (walking, standing up, lying on the ground, crouching, moving down, moving up, sitting, lying on a sofa) and fall action in the dataset. The fall action happens in 22 scenarios, and the other two only contain confounding actions. We can find the beginning and ending frame index of the fall actions from a technical report [3].

Le2i [7] is a simulated fall detection dataset. It contains 156 fall videos and 65 ADL (e.g., walking, sitting down, squatting, etc.) videos. All the videos are simulated by actors and captured from four scenarios (home, coffee room, office, and lecture room). It should be noted that this dataset includes some detection challenges, such as the occlusions, illumination change, fake fall, and so on. Because the official label is incomplete, we made detailed frame-level labels for all the videos.

URFD [15] is a simulated fall detection dataset with the RGB-Depth information. It contains 30 fall videos and 40 ADL videos, and several Microsoft Kinect sensors record them. We do not use the depth channel in our experiments. The dataset includes the start and end time in seconds for the fall action. To train the frame-level algorithm pipeline, we made a more specific label at the frame-level for this dataset.

The **High Quality Fall Simulation (HQFS)** [4] is a new fall detection dataset from the realistic surveillance video. It contains

Table 2: Five public available fall detection datasets. To support the needs of algorithm training and testing, we made detailed labels for these datasets, as described in section 4.1.

Dataset Name	Video Type and Number
MCFD [3]	176 Fall + 16 ADL
Le2i [7]	156 Fall + 65 ADL
URFD [15]	30 Fall + 40 ADL
HQFS [4]	274 Fall + 17 ADL
YTBF [13]	88 Fall + 0 ADL

Table 3: Compare pose motion feature on the Le2i dataset.

Feature	Task	Precision	Recall	F1 score
OpenPose	FLFD	0.9487	0.9427	0.9457
PoseFlow		0.9615	0.9615	0.9615
DI	SLFD	0.7346	0.7625	0.7483
PoTion		0.6961	0.7789	0.7352
DPM		0.7521	0.8014	0.7760
NPF		0.7939	0.8397	0.8161

274 fall videos and 17 ADL videos, and all the filmed scenarios is similar to a nursing room. Different from the simulated datasets, it is close to the real-life situation and includes much more detection challenges, such as the person will go out of the camera view. To evaluate the proposed methods and benchmark method [13], we also made detailed frame-level labels for this dataset.

The **YouTube Fall (YTBF)** [13] dataset is an entirely real-life dataset. It is collected from YouTube by searching keywords such as fall, trip, slip, topple, tumble, and so on. It contains 88 fall videos, and there are a total of 430 fall actions from these videos. This dataset includes both indoor and outdoor scenarios and is very challenging. Most of the videos are shot by mobile phone and shake violently. Because this dataset does not involve the frame-level labels, we only made the video-level fall detection on it.

4.2 Experiment Settings

Because the fall action usually happens in a short time, the positive and negative samples are quite unbalanced in the fall detection problem, which will cause the model hard to train. To tackle this problem, we augment the fall frames to three times by the random cropping and horizontally flipping, what operations widely accepted for other computer vision tasks. However, we do not flip any frames in vertical direction since most of the fall actions are from a high place to a lower position in the video.

We use different evaluation metrics for the multi-level tasks. For the FLFD and SLFD, we calculate the *Precision*, *Recall* and *F1 score* for every input, and the formulation is $F_1 = 2 * (percision * recall)/(percision + recall)$ in detail. For the VLFD task, we detect the start index and the end index, and calculate the intersection-over-union (*IOU*) between the true index and the predicted index, which is $IOU = \frac{[predict\_start, predict\_end]\cap[start\_label, end\_label]}{[predict\_start, predict\_end]\cup[start\_label, end\_label]}$.

Table 4: Compare SLFD baselines (*SLAP, NC_2017*[13]).

Dataset	Method	Precision	Recall	F1 score
Le2i	NC_2017	0.9359	0.9419	0.9389
URFD		0.8333	0.8333	0.8333
YTBF		0.6250	-	-
MCFD	Proposed SLAP	0.9602	0.9548	0.9575
Le2i		0.9487	0.9427	0.9457
URFD		0.8667	0.8387	0.8525
HQFS		0.8102	0.9652	0.8810
YTBF		0.6591	-	-

Table 5: For the VLFD task, we use the same SLAP backbone with different sampling strategies.

Dataset	Sampling Strategy	Avg. IOU
Le2i	Sliding Window (window length $k = 60$)	42.72%
	Shot-based Sampling (VLFD pre-trained)	43.17%
HQFS	Sliding Window (window length $k = 60$)	31.55%
	Shot-based Sampling (VLFD pre-trained)	33.18%
YTBF	Sliding Window (window length $k = 60$)	17.72%
	Shot-based Sampling (VLFD pre-trained)	18.94%

4.3 Ablation Study for Pose Motion Feature

To evaluate the influences of different pose estimators and motion representations, we conduct the ablation study on the SLFD task. We use the OpenPose[6] and PoseFlow [29] to generate different pose heatmaps and test them respectively. To evaluate the influences of different pose motion representations, we also compared the *dynamic image (DI)*[5], *PoTion*[8], the proposed *Dynamic Pose Motion (DPM)*, and the *normalized partial feature (NPF)*, respectively. The results are shown in table 3 and similar relative performance are observed on other datasets. To make a fair comparison and avoid the numerical influence, we make the normalizing operation on the feature map of dynamic image, PoTion, DPM, and NPF, respectively.

4.4 Results on FLFD, SLFD and VLFD

To do the FLFD task, we first extract the pose heatmaps from the pre-trained Openpose COCO model. Then we train a 4-layer (28×14×14×1) MLP for the fall prediction. We make the FLFD experiments on all the datasets. To do the SLFD task, we use the proposed algorithm pipeline, as stated in section 3.2 and figure 3. For all the above-mentioned experiments, we adopt the same pre-trained VGG model as used in [13], and fine-tune a 100-hidden-units' LSTM for different fall datasets. The results are shown in table 4.

Although from the FLFD task, the Poseflow is demonstrated a better pose feature, for the reason of implementation, we still use the Openpose COCO model to extract the pose heatmap in the SLFD and VLFD tasks. When the number of ADL videos are too few than the fall videos, the F1 score will be deceptive. Therefore, for the SLFD task, we make the experiments on the augmented MCFD and HQFS datasets.

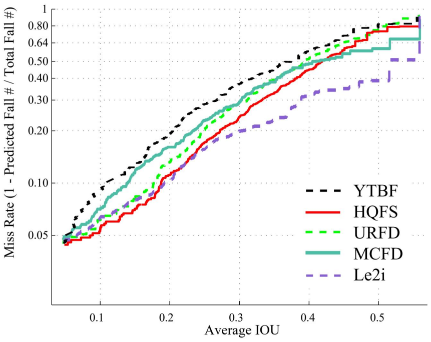

Figure 5: The miss rate curves of the proposed VLAP.

For the video-level algorithm pipeline (VLAP), we implement the *sliding window* and *shot-based* sampling strategies on all videos. We randomly choose 80% videos to be training set and leave the others to be the testing set. In order to train our method across different datasets, we modify all the videos with 60 fps. We record the IOU performance among the testing videos and report the average IOU in table 5. We also show the missing rate curve for different datasets, as shown in figure 5. As missing a fall action may lead to serious consequences, we can increase the threshold of average IOU value in the real-life applications of VLAP.

5 CONCLUSION

In this paper, we propose a multi-level definition for the fall recognition from ADL. The different task has a different objective, thus we can focus on different feature representation or develop a suitable algorithm pipeline for one specific task. For the SLFD task, we propose a novel DPM representation and the corresponding algorithm pipeline. The proposed SLAP establishes a new baseline and achieve a better performance than current method, on both of the simulated and real-life datasets.

6 ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under Grant 61771273, the RD Program of Shenzhen under Grant JCYJ20180508152204044, and the Research Fund of PCL Future Regional Network Facilities for Large-scale Experiments and Applications (PCL2018KP001).

REFERENCES

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7297–7306.

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*. 5803–5812.

[3] Edouard Auvinet, Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. 2010. Multiple cameras fall dataset. *DIRO-Université de Montréal, Tech. Rep* 1350 (2010).

[4] Greet Baldewijns, Glen Debard, Gert Mertes, Bart Vanrumste, and Tom Croonenborghs. 2016. Bridging the gap between real-life data and simulated data by providing a highly realistic fall dataset for evaluating camera-based fall detection algorithms. *Healthcare technology letters* 3, 1 (2016), 6–11.

[5] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. 2016. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3034–3042.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7291–7299.

[7] Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. 2012. Definition and performance evaluation of a robust SVM based fall detection solution. In *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 218–224.

[8] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. 2018. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7024–7033.

[9] Jia-Luen Chua, Yoong Choon Chang, and Wee Keong Lim. 2015. A simple vision-based fall detection technique for indoor video surveillance. *Signal, Image and Video Processing* 9, 3 (2015), 623–633.

[10] Qianggang Ding, Fan Yang, Jiawei Li, Sifan Wu, Bowen Zhao, Zhi Wang, and Shu-Tao Xia. 2019. RT-ADI: Fast Real-Time Video Representation for Multi-view Human Fall Detection. In *2019 IEEE International Conference on Multimedia & Expo Workshops*. IEEE, 13–18.

[11] Giovanni Diraco, Alessandro Leone, and Pietro Siciliano. 2010. An active vision system for fall detection and posture recognition in elderly healthcare. In *Proceedings of the conference on design, automation and test in Europe*. European Design and Automation Association, 1536–1541.

[12] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2625–2634.

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

[14] Zhanyuan Huang, Yang Liu, Yajun Fang, and Berthold KP Horn. 2018. Video-based Fall Detection for Seniors with Human Pose Estimation. In *4th International Conference on Universal Village*. IEEE, 1–4.

[15] Bogdan Kwolek and Michal Kepski. 2014. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine* 117, 3 (2014), 489–501.

[16] Bogdan Kwolek and Michal Kepski. 2015. Improving fall detection by the use of depth sensor and accelerometer. *Neurocomputing* 168 (2015), 637–645.

[17] Inwoong Lee, Doyoung Kim, Seungyoon Kang, and Sanghoon Lee. 2017. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1012–1020.

[18] Muhammad Mubashir, Ling Shao, and Luke Seed. 2013. A survey on fall detection: Principles and approaches. *Neurocomputing* 100 (2013), 144–152.

[19] Adrián Núñez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. 2017. Vision-based fall detection with convolutional neural networks. *Wireless communications and mobile computing* 2017 (2017).

[20] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. 2007. Fall detection from human shape and motion history using video surveillance. In *21st International Conference on Advanced Information Networking and Applications Workshops*, Vol. 2. IEEE, 875–880.

[21] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*. 568–576.

[22] Markus D Solbach and John K Tsotsos. 2017. Vision-based fallen person detection for the elderly. In *Proceedings of the IEEE International Conference on Computer Vision*. 1433–1442.

[23] Erik E Stone and Marjorie Skubic. 2015. Fall detection in homes of older adults using the Microsoft Kinect. *IEEE journal of biomedical and health informatics* 19, 1 (2015), 290–301.

[24] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[25] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. 2018. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1430–1439.

[26] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. 2017. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4325–4334.

[27] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. 20–36.

[28] WHO. 2008. *WHO global report on falls prevention in older age.* World Health Organization and World Health Organization. Ageing and Life Course Unit.

[29] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. 2018. Pose Flow: Efficient Online Pose Tracking. In *The British Machine Vision Conference.*

[30] Yan Zhang and Heiko Neumann. 2018. An empirical study towards understanding how deep convolutional nets recognize falls. In *Proceedings of the European Conference on Computer Vision.*