An Empirical Study Towards Understanding How Deep Convolutional Nets Recognize Falls

Yan Zhang(✉) and Heiko Neumann

Institute of Neural Information Processing, Ulm University, Ulm, Germany
{yan.zhang,heiko.neumann}@uni-ulm.de

Abstract. Detecting unintended falls is essential for ambient intelligence and healthcare of elderly people living alone. In recent years, deep convolutional nets are widely used in human action analysis, based on which a number of fall detection methods have been proposed. Despite their highly effective performances, the behaviors of how the convolutional nets recognize falls are still not clear. In this paper, instead of proposing a novel approach, we perform a systematical empirical study, attempting to investigate the underlying fall recognition process. We propose four tasks to investigate, which involve five types of input modalities, seven net instances and different training samples. The obtained quantitative and qualitative results reveal the patterns that the nets tend to learn, and several factors that can heavily influence the performances on fall recognition. We expect that our conclusions are favorable to proposing better deep learning solutions to fall detection systems.

Keywords: Deep convolutional nets · Fall recognition
Empirical study

1 Introduction

Due to cognitive impairment or deficiencies of motor functionalities, unintended falls occur frequently in the group of elderly people, and can lead to severe or even fatal injuries [8,9]. Therefore, to build up fall detection systems for elderly people healthcare, it is essential to recognize falls in an automatic and effective manner.

Fall recognition has been intensively studied in the past. If the human body dynamics has been precisely measured, identifying an unintended fall is straightforward. For example, one can recognize falls via measuring the vertical velocity of the human body towards the ground. If the velocity is above a threshold, then a fall occurs. Consequently, researchers tend to propose novel solutions to capture the body configurations and motions. For example, the work of [44] uses a wearable triaxial accelerometer to measure the body motion and recognizes falls via one-class support vector machine. The work of [41] develops a wearable

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11134, pp. 112–127, 2019.
https://doi.org/10.1007/978-3-030-11024-6_8

system (mainly based on the accelerometer and GPS) to detect and localize falls in the wild. Wearable sensors enable measuring physical attributes of the human body in a precise and real-time manner. However, the sensors have to be physically attached to people, causing obstructive interventions to their daily living activities.

Computer vision technologies realize non-obstructive measurement of human body motions and conduct behavior recognition only based on imagery data. The effectiveness is highly improved when deep convolutional networks trained on large-scale image datasets are employed. To recognize a fall, two families of methods can be considered: The first family attempts to capture precise body configurations over time, such as [4] and [14] for 2D pose estimation and [12] for 3D pose estimation. Such pose estimation methods can replace the functionality of wearable sensors but perform human body measurement in a non-contacting manner. The second family, which is usually based on deep convolutional nets, aims at inferring the semantic content of the input data via creating a mapping from the input data to the action labels in an end-to-end fashion. For example, [31] yields an action label for an input sequence, [23] yields both action labels and temporal durations, and [17] produces frame-wise labels for temporal action segmentation. In this paper, we focus on the second family of methods, since the end-to-end inference behavior does not need any intermediate step, e.g., training a classifier based on the captured body poses. In addition, the data annotation procedure only requires to assign action labels to frames/videos, instead of annotating the key joints on the human bodies in each frame as the first family of methods.

Although several relevant methods like [24] have been proposed, the underlying reasons of the effectiveness are still not clear. In this paper, rather than proposing a novel method for fall recognition, we aim at attaining insights of how the deep convolutional net recognizes falls via a series of empirical investigations. Our study is based on a family of convolutional encoder-decoder nets, different types of input modalities and recordings from different environments. According to our investigations, we discover: (1) Human body motion represented by the optical flow is highly informative for the net to recognize falls. (2) The net tends to learn human body-centered context, namely the appearance surrounding the human body, if the training samples have RGB frames. However, the net cannot get rid of the influence of the background context irrelevant to falling, and lacks generalizability across different environments. (3) The human-centered context and human body motion are complementary. (4) Inaccurate body pose information can degrade the performances.

Organization. This paper is organized as follows. Section 2 introduces related work on vision-based methods for fall recognition and work on model explanation. Section 3 introduces the employed convolutional net, as well as different sorts of attribute maps for model explanation. In Sect. 4, we present our empirical investigations, results and discussions. In the end, we conclude our work and propose future studies in Sect. 5.

2 Related Work

Systematic reviews of fall recognition and detection systems can be found in [13] and [22], which cover solutions based on diverse types of sensors. For vision-based methods, a typical processing pipeline consists of background subtraction, feature extraction and classification, as presented in [27,39] and others. Each step in this pipeline is normally hand-crafted, separately considered and implemented based on certain heuristic rules. A frequently considered rule is that the background information is redundant for fall detection. Thus, background subtraction is performed by algorithms like training Gaussian mixture models, subspace clustering or other sophisticated approaches [25]. Another heuristic rule is that the body shape is a pronounced feature of falling. Consequently, the silhouette of the human body [2,27], or the shape of the foreground bounding box [38,39], is extracted and analyzed. Nevertheless, heuristics are not always precise and comprehensive. The studies of [35] and [34] present effective fall detection solutions when considering the ground plane, indicating that the background information can be very useful.

Comparing with traditional vision-based approaches, deep learning methods enable end-to-end inference with minimal pre-processing on the input data, and the deep nets can learn representative features from the data automatically. Therefore, the algorithm is not necessary to rely on non-guaranteed heuristics. Several studies report that deep learning methods lead to better performances in terms of action recognition [5,31], action detection [10,28,43], action parsing [17,19] and other tasks of human behavior analysis. Their success encourages many studies of fall recognition based on deep neural networks. For example, the work of [24] employs a convolutional net with a similar architecture to the VGG-16 net [32] and uses optical flow as the input modality. The work of [40] uses a PCANet to recognize falls from image sequences with the assistance of foreground detection.

To understand the behaviors of deep convolutional nets, several types of attribute maps have been proposed [1,3]. For a specific input and a target class, the attribute map has the same spatial resolution with the input, and reveals the influence of each input pixel to the probability of the target class. The work of [30] proposes a saliency map, which is computed as the derivative of the output with respect to the input. [36] proposes the integrated gradients, in which the values show the difference between the net output of a reference input (normally zero) and the net output of a sample. [29] proposes the DeepLIFT attribute measure, which can be regarded as an approximated version of integrated gradients according to [1].

3 The Convolutional Net

We formulate fall recognition as a binary classification problem, and expect to obtain frame-wise semantic labels, so that recognition and temporal localization can be solved simultaneously. Therefore, we use a convolutional encoder-decoder

(CED) architecture, which is modified from the non-causal ED-TCN model [17]. Comparing with [17], our CED net combines the spatial net and the temporal net into a coherent structure. The architecture is illustrated in Fig. 1 and the specifications are presented in Fig. 2.

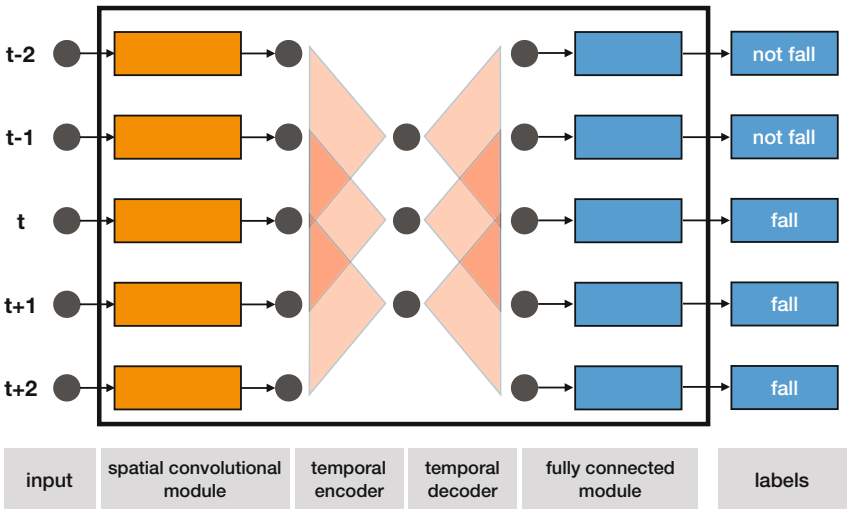

Fig. 1. The architecture of the convolutional encoder-decoder (CED) net. In the input layer, each frame (the gray node) is a 3D tensor with [height, width, channels].

The CED architecture has several advantages besides generating frame-wise labels: (1) The CED net can capture long-range temporal dependencies, and outperforms recurrent nets, e.g. bidirectional LSTMs [11,33], w.r.t. temporal action segmentation [17] and motion prediction [20]. (2) The CED net can generate piece-wise constant label sequences directly, without post-processing steps like median filtering. (3) Comparing with recurrent neural nets, in our trials we find that CED is much easier to train and converges much faster. (4) Once CED is trained, the model can process sequences of arbitrary lengths. Because of such merits, we only consider the convolutional net in our study, and investigating recurrent neural nets is beyond our scope.

The CED model consists of several modules as shown in Fig. 1. In the following content, we introduce each of them.

3.1 The Spatial Convolutional Module

Our convolutional module aims at extracting the feature of each individual frame in the video. It consists of three convolutional blocks, and each block contains a 2D convolutional layer, an activation function layer and a 2D max-pooling

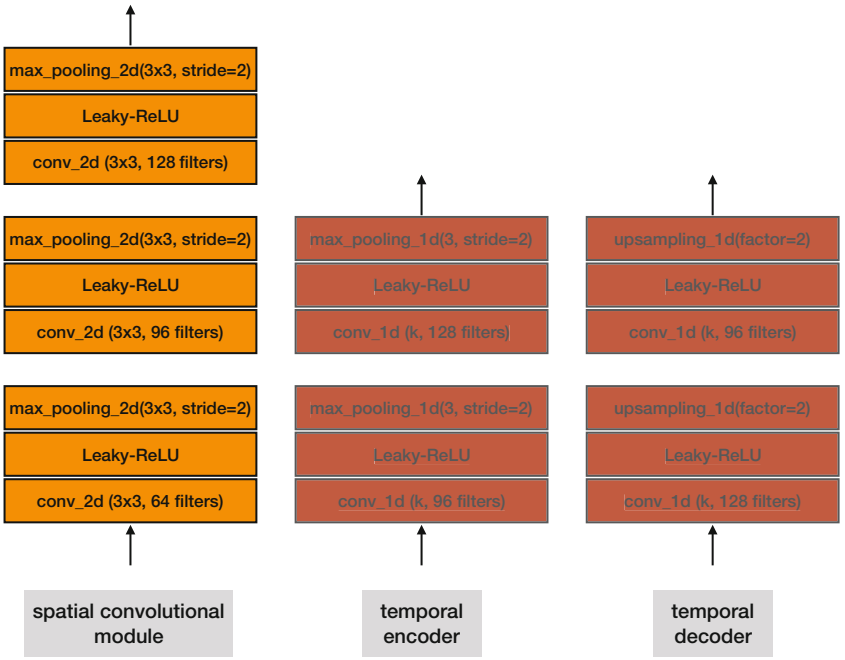

Fig. 2. Module specifications of our CED net, in which the data flows from the bottom to the top. The value of k is determined in Sect. 4.

layer, following the architecture of the VGG net [17,32]. After each block, the spatial resolution is downsampled by the factor of 2. At the end of the module, the input 3D tensor is flattened to a 1D vector. The specifications of the spatial convolutional module are shown in Fig. 2. The number of convolution filters are suggested by [18]. In our work, we use the leaky-ReLU [21] activation function, due to the superior performances to standard to the ReLU function, as indicated in [42]. Moreover, the spatial convolution module is applied on each individual frame of the input tensor sequence, and has shared parameters across all frames.

3.2 The Temporal Encoder and Decoder

After the spatial convolutional module, the 3D tensor of each frame converts to a 1D vector, and then all the vectors are stacked along the temporal dimension to compose a 2D tensor with the shape of [time, dimension] (or a 3D tensor with the shape of [batch, time, dimension]).

Similar to the 2D convolution operation, which can effectively capture spatial local features, the 1D temporal convolution computes temporal correlations between frames, in which the value of the kernel size k specifies the size of the receptive field. The 1D max pooling operation downsamples the data along the time dimension to yield a compressed data representation. On the other and,

the upsampling operation increases the temporal resolution to recover the original time length. The encoder and decoder have symmetric architectures, and hence the temporal encoder input and the temporal decoder output has the same temporal length.

3.3 The Fully Connected Module

The fully connected module consists of a fully connected layer, a dropout layer and a softmax layer, and is applied on individual frames in the output of temporal decoder with shared parameters. Due to our binary classification setting, the output dimension of the fully connected layer can be 1 or 2. Here we use the two-dimensional output, since we expect that the insights derived from our work can be extended to multi-class classification problems straightforwardly. The dropout layer (with a keep ratio of 0.5) is used to avoid overfitting, and the softmax layer converts the scores to probabilities.

3.4 Training the Network

In our work, all the modules are trained jointly, in contrast to [17] that only trains the temporal encoder-decoder using the outputs from a pre-trained spatial net. For each frame, we compute the *cross-entropy* between the one-hot encoded ground truth label and the softmax output. Then the loss of the sequence is the sum of the cross-entropy values of all frames. After specifying the loss, the model parameters are learned via the Adam algorithm [16]. Comparing with the stochastic gradient descent method, Adam can lead to superior results as reported in [16]. In addition, the adaptive momentum nature is suitable for our problems, since our input modality can cause sparse gradients, like optical flows with motion information only on the foreground. Implementation details refer to Sect. 4.

4 Experiments

In this section, we present our empirical experiments to investigate how the deep convolutional net CED recognize falls. We propose 4 tasks, and for each task the quantitative results are shown by cross-validated frame-wise accuracies and the qualitative results are shown by attribute maps.

4.1 Dataset

We use the **Le2i** Fall detection dataset presented in [7], which is built using a single camera in realistic surveillance setting containing illumination variations, occlusions by furnitures, different appearances of the subjects, different types of falls (e.g. falling forward, falling while sitting, etc.), and other factors that simulate falls in daily lives. The video has spatial resolution of 320×240 of

pixels and is captured with 25 fps. Each video is annotated in a frame-wise fashion, which fits the CED architecture.

The original dataset contains 4 environments, i.e. *home*, *lecture room*, *coffee room* and *office*. Due to loss of annotation files, we only use the recordings from *home* and *coffee room* in our study. For each of the two environments, there exist two groups of recordings.

Fig. 3. From left to right: (1) Sample frames of falls in *home* and *coffee room*. (2) The statistics of time durations of falls across all videos, in which the x-axis denotes the fall duration, the y-axis and the bins show the normalized occurrence frequencies and the curve shows the fitted distribution.

Data Preparation. Since we focus on frame-wise fall recognition, in order to balance the number of fall and not-fall frames, from each video containing falling we extract a video snippet consisting of frames before, during and after the fall. Video trimming is based on the statistics of time durations of falls, which is shown in Fig. 3. Specifically, the extracted snippet has 60 frames (2.4 s), starting from $T - 49$ to $T + 10$, where T is the timestamp of the last frame of fall in the video.

Depending on the recording environment, we perform a *high-level* splitting to divide the dataset into 2 folds, each of which contains recordings from either *home* or *coffee room*. Since there are two groups for each environment, we perform a *low-level* splitting to divide the dataset into 4 folds. Therefore, the *high-level* splitting can be used for cross-environment validation, and the *low-level* splitting can be used for cross-validation under small environment variations.

After such preparation step, we obtain a new dataset incorporating 99 video snippets with 2 *high-level* splits and 4 *mid-level* splits.

4.2 Input Modalities to the Net

Besides the RGB frames, we also compute time differences, TV-L1 optical flows [6], and score maps of human body poses[1] [14,15] as the net input modalities. For computational purposes, we downsample the spatial resolution to 56×56.

[1] The MPII body model has 14 keypoints and hence the method generates 14 pose score maps for each image. In our experiment, we average these 14 score maps to one map.

Similar to [17], each frame of the net input contains a stack of frames from the original data sequence. Denoting the *standardized* RGB image sequence as $\{I_t\}$, the optical flow sequence as $\{w_t\}$ (values within $[-20, 20]$) and the sequence of score maps as $\{s_t\}$ (values within $[0, 1]$), the modalities used in our experiments are shown in Table 1.

Table 1. The input modalities used in our experiments, in which the **Pose+Optical Flow** modality uses the normalized optical flow $\tilde{w}_t$.

Modalities	Format of each frame
RGB+TimeDifference	$\{I_{t-1}, I_t, I_{t+1}, I_t - I_{t-1}, I_{t+1} - I_t\}$
TimeDifference	$\{I_t - I_{t-1}, I_{t+1} - I_t\}$
Optical Flow	$\{w_{t-1}, w_t, w_{t+1}\}$
Pose	$\{s_{t-1}, s_t, s_{t+1}\}$
Pose+Optical Flow	$\{s_{t-1}, \tilde{w}_{t-1}, s_t, \tilde{w}_t, s_{t+1}, \tilde{w}_{t+1}\}$

The **RGB+TimeDifference** modality is suggested by [17], in which the RGB frames encode the appearances of the visual scene and the time differences have the functionality of attention. Image standardization is performed frame-wisely, in order to eliminate the influence of illumination changes. Since the background is static, **TimeDifference** and **Optical Flow** focus on the human body, while **TimeDifference** does not incorporate directional human body motions. The pose information is represented by the score map produced by the pre-trained model of [14,15], which is beneficial for person re-identification and tracking [37]. When combining optical flow and pose, we expect that the pose score map works as an attention mechanism, encouraging the net to learn motion features around the body key points.

One can note that the **Pose+Optical Flow** modality uses the normalized optical flow $\tilde{w}_t$, which is computed by $w_t/20$ and hence ranges within $[-1, 1]$. In this case, the ranges of the pose score map and the optical flow are similar. We find that such flow normalization process is beneficial in our trials. A probable reason is that the normalization leads to similar ranges of convolution parameters for the flow and the pose map in **Pose+Optical Flow**.

4.3 Implementation

The implementation is based on Tensorflow. The batch size is fixed to 8, meaning 8 tensor sequences are fed to the net for one iteration. The Adam algorithm is used to train the model [16], where the initial learning rate is 0.001 and other parameters are set to the Tensorflow default values. The learning rate is decayed every 10 epochs, namely $0.001 \times 0.9^{\lfloor \frac{epoch}{10} \rfloor}$, and training terminates after 100 epochs. In our trials, more iterations lead to comparable or worse results.

In addition, attribute map extraction is implemented based on the DeepExplain library introduced in [1].

4.4 Evaluation Methods

Rather than performing model selection as in [17], we use a family of net instances to verify whether some conditions can consistently influence the performances. We vary two influential factors in the net architecture, i.e., the temporal convolution kernel size k determining the temporal receptive field, and the temporal length of the input sequence l determining the up-limit range of the temporal structure that the net can learn. In our experiments, we use the net instances with $(k, l) \in \{(3, 8), (3, 16), (3, 32), (5, 16), (5, 32), (7, 16), (7, 32)\}$.

For the *high-level* splitting, 2-fold cross-validation is performed, in which each net instance is trained on the first fold and validated on the second, and vice versa. Then, for each net instance, the two validated accuracies are averaged to derive the cross-validated accuracy. For the *low-level* splitting, 4-fold cross-validation is performed in an identical manner. Since each net instance associates with an accuracy value, the quantitative performance of the CED model is presented in terms of a box plot.

The qualitative results are shown by attribute maps, i.e. DeepLIFT [29], integrated gradients [1] and saliency maps [30]. In addition, each attribute map is stacked to the map of edges of the input for visualization purposes.

4.5 Tasks, Results and Discussions

Task 1: Investigating the Cross-Environment Generalizability. In this task, we aim at investigating the generalizability across environments, namely, how the CED performs if training samples and testing samples are collected from totally different environments. Therefore, we conduct a 2-fold cross-validation procedure based on the *high-level* splitting, and use **RGB+TimeDifference**, **TimeDifference** and **Optical Flow** as the input modalities. The results are shown in Fig. 4.

From the box plots, one can see that **RGB+TimeDifference** performs inferior to **TimeDifference** and **Optical Flow**, and **Optical Flow** outperforms **TimeDifference**. In addition, the attribute maps from four testing recordings consistently show that many pixels on the background can heavily affect the net inference process.

Discussion. The net with **RGB+TimeDifference** performs just slightly better than random guess, due to the binary classification setting. The attribute maps show that irrelevant background information has strong influence on fall recognition, and hence we consider that the net cannot discard irrelevant background information automatically during training, and leads to degraded generalizability across environments. Excluding the background information, as in **TimeDifference** and **Optical Flow**, can improve the performances dramatically. This fact can indicate that real influential and environment-invariant features of falls are human body-centered. In addition, the superior performances of **Optical Flow** to **TimeDifference** can indicate that the directional body motion contains more representative information of falls.

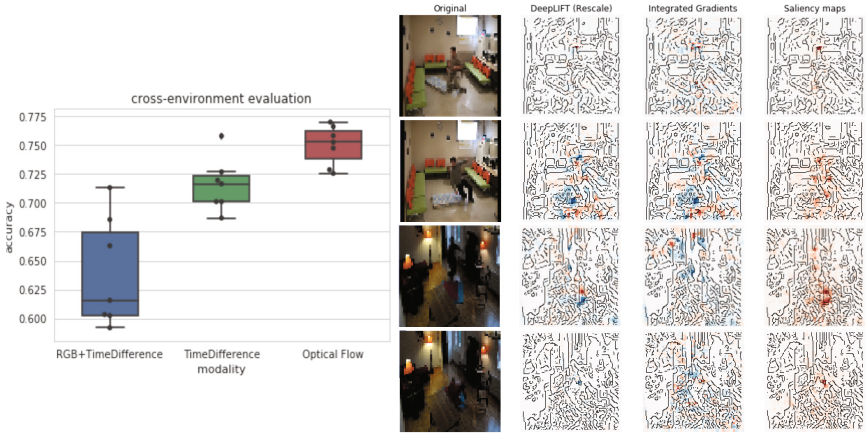

Fig. 4. From left to right: (1) The quantitative results of the 2-fold cross-validation, where the results from each net instance are shown as black dots in parallel to the box plots. In each box plot, the bar inside the box denotes the median, and the box shows the interquartile range (IQR) and the samples between whiskers with 1.5×IQR are inliers. (2) The attribute maps of frames from four testing recordings are shown, where the red color and the blue color denote contribution and suppression effects on the probability of falling. (Color figure online)

Task 2: Investigating the Influence of Training Samples. In this task, we aim at investigating the influence of training samples recorded from similar environments to the testing samples. Thus, we perform 4-fold cross-validation based on the *low-level* dataset splitting, and compare the performances with the 2-fold cross-validation setting (see Task 1). The employed input modality is **RGB+TimeDifference** and the results are shown in Fig. 5. The reason of only using **RGB+TimeDifference** is that other modalities used in Task 1, namely, **TimeDifference** and **Optical Flow**, are environment-independent and cannot reveal the influence of environment variations.

The box plots show that the training recordings from similar environments to testing can largely improve the performances. Indicated by all attribute maps on the right, we can find that the influential pixels noticeably become more human body-centered.

Discussion. Quantitatively, training recordings similar to the testing recordings are highly favorable. The reason can be revealed from the attribute maps. Specifically, the fact that influential pixels are more concentrated around the human body can also indicate that the fall features are human body-centered. In addition, one can notice that the body-centered influential pixels tend to locate around the contour of the body, instead of directly on the body. This fact may indicate that the body-centered context, or the interaction between the human body and the environment, is a representative feature of fall.

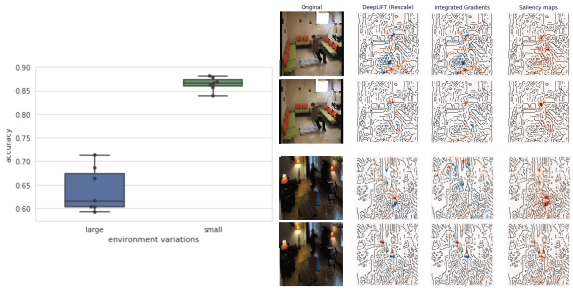

Fig. 5. From left to right: (1) The quantitative results under large environment variations (the *high-level* splits) and small environment variations (the *low-level* splits), with the modality **RGB+TimeDifference**. (2) Attribute maps from two testing samples are shown. The first two rows compare the large and small evaluation settings on the same frame in *coffee room*, respectively. The last two rows show another comparison on the same frame in *home*.

Task 3: Investigating the Human Body-Centered Pattern. Based on the results in Task 1 and Task 2, we believe that the convolutional net tends to learn body-centered patterns for fall recognition. Here we perform further investigations based on the *low-level* data splitting and the **RGB+TimeDifference** and **Optical Flow** modalities, which represent body-centered context and body motion, respectively. Afterwards, we fuse the two modalities following the work of [31]. Specifically, we average the softmax outputs from two streams of CED nets with the same (k, l) values. Figure 6 shows the results.

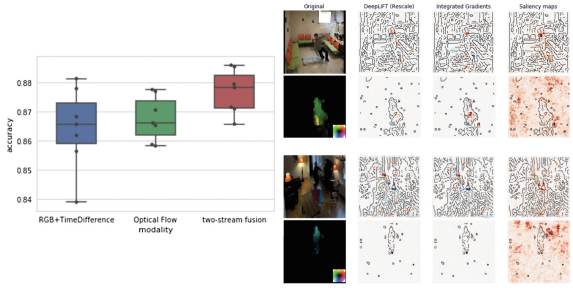

Fig. 6. From left to right: (1) The quantitative results of different modalities under small environment variations (the *low-level* splits). (2) Examples of attribute maps of the two modalities are presented. In particular, the optical flow is visualized using the color coding scheme attached at the bottom-right corner. (Color figure online)

One can see that **Optical Flow** and **RGB+TimeDifference** lead to comparable performances according to the box plots, yet the net with the **Optical Flow** modality behaves more stable than the other case. The fusion results outperform individual modalities. Additionally, from the attribute maps of optical

flows we can see that the influential pixels are within the contour of the human body, in contrast to the attribute maps of **RGB+TimeDifference**. One can note that the saliency map is not reliable for **Optical Flow**, since the saliency values are computed as the derivatives of the output w.r.t. the input and zero-value input can cause numerical problems.

Discussion. A probable reason of the stable performance with **Optical Flow** is that human body motion can represent falls more robustly than the body-centered context, which can be easily influenced by the background information. The superior performance of modality fusion can indicate that body-centered context and body motion are complementary. The complementary property can also be viewed from the attribute maps, since the influential pixels are at different locations.

Task 4: Investigating the Influence of Body Pose Information. Here we aim at investigating the influence of the 2D pose information. Since motion capture devices are not used in the dataset and no body pose annotations are available, the pose maps are extracted using the pre-trained model associated with [14,15]. The evaluation is based on the *low-level* splitting, as well as the modalities of **Pose**, **Optical Flow** and **Pose+Optical Flow**. The results are shown in Fig. 7.

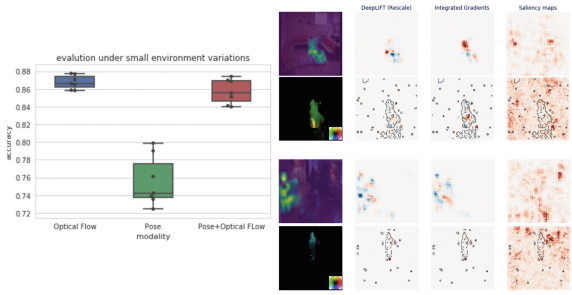

Fig. 7. From left to right: (1) The quantitative results presented by box plots. (2) The attribute maps of pose and optical flow modalities. The selected frames are the same with previous figures. The pose score map, in which the value increases from blue to yellow, is overlaid with the RGB image only for visualization. The RGB image is not input to the net. (Color figure online)

One can see that the pose information leads to inferior performances, and also deteriorates the performances of **Optical Flow** when combing flow and pose information. On the right hand, one can find that the influential pixels on the pose score maps mainly locate at the positions the non-zero pose scores. Similar to the optical flow case, the saliency maps of **Pose** are deteriorated by numerical problems. Moreover, from the third row on the right, one can see that the pose estimation is not always reliable.

Discussion. Pose estimation from images is a challenging problem. Although the state-of-the-art algorithms perform quite well on standard benchmarks, the estimation result is not guaranteed. In case of fall recognition, we can see that incorrect pose estimation can dramatically degrade the performances.

5 Conclusion and Future Work

In this paper, we aim at investigating the behaviors of the convolutional neural net when conducting fall recognition. To enable frame-wise recognition, we use the convolutional encoder-decoder (CED) architecture and employ a set of net instances. Based on different types of input modalities and dataset splits, our empirical studies show several influential factors of the model performances. In particular, we find that: (1) The net tends to learn body-centered patterns, but cannot eliminate the influence of background information, leading to poor cross-environment generalizability. Therefore, for cross-environment uses in practice, it is better to perform person detection as a pre-processing step, or incorporate a region-of-interest proposing module into an end-to-end model, like the Faster R-CNN model [26]. (2) Training samples captured from the testing environment can considerably improve the performance and encourage the net to encode body-centered context, for which the most influential pixels are located around the body contour. Thus, in practice, we suggest to collect training samples from the deployment environment when possible. (3) The human body motion contains representative features of falls robust to environment changes, and influences on fall recognition in a complementary manner with the body-centered context. In this case, we suggest to use the two-stream (the appearance stream and the motion stream) architecture [31] when detecting falls. In addition, since the body-centered context and the body motion are from different image regions, their correlation could be trivial and we probably can effectively fuse the two types of feature vectors only by concatenation or averaging. (4) Incorrect pose information can degrade the performances heavily. At the current stage, body pose estimation is a challaging task by itself, and the performances are not guaranted. We hence recommend not to incorporate pose information for fall recognition without additional checking.

Herein we focus on trimmed videos for investigating the net behaviors. Based on the obtained insights, we consider to develop an effective fall detection system based on the CED architecture for untrimmed videos or even streaming data in future.

Acknowledgements. This work is supported by a grant of the Federal Ministry of Education and Research of Germany (BMBF) for the project of SenseEmotion.

References

1. Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: A unified view of gradient-based attribution methods for deep neural networks. arXiv preprint arXiv:1711.06104 (2017)
2. Anderson, D., Keller, J.M., Skubic, M., Chen, X., He, Z.: Recognizing falls from silhouettes. In: Proceedings of the 28th IEEE EMBS Annual International Conference, pp. 6388–6391. IEEE (2006)
3. Babiker, H.K.B., Goebel, R.: An introduction to deep visual explanation. arXiv preprint arXiv:1711.09482 (2017)
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
5. Carreira, J., Zisserman, A.: Quo Vadis, action recognition? A new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
6. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imag. Vis. **20**(1–2), 89–97 (2004)
7. Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R.: Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. J. Electron. Imag. **22**(4), 041106 (2013)
8. Dykes, P.C., et al.: Fall prevention in acute care hospitals: a randomized trial. Jama **304**(17), 1912–1918 (2010)
9. Gillain, S., Elbouz, L., Beaudart, C., Bruyère, O., Reginster, J., Petermans, J.: Falls in the elderly. Revue medicale de Liege **69**(5–6), 258–264 (2014)
10. Gkioxari, G., Malik, J.: Finding action tubes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 759–768. IEEE (2015)
11. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 799–804. Springer, Heidelberg (2005). https://doi.org/10.1007/11550907_126
12. Güler, R.A., Neverova, N., Kokkinos, I.: Densepose: dense human pose estimation in the wild. arXiv preprint arXiv:1802.00434 (2018)
13. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. Biomed. Eng. Online **12**(1), 66 (2013)
14. Insafutdinov, E., et al.: Arttrack: articulated multi-person tracking in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1293–1301 (2017)
15. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 34–50. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_3
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1003–1012, July 2017
18. Lea, C., Reiter, A., Vidal, R., Hager, G.D.: Segmental spatiotemporal CNNs for fine-grained action segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 36–52. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_3

19. Lea, C., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks: a unified approach to action segmentation. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 47–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_7
20. Li, C., Zhang, Z., Lee, W.S., Lee, G.H.: Convolutional sequence to sequence model for human dynamics. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5226–5234 (2018)
21. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of ICML, vol. 30, p. 3 (2013)
22. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: principles and approaches. Neurocomputing **100**, 144–152 (2013)
23. Neverova, N., Wolf, C., Taylor, G.W., Nebout, F.: Multi-scale deep learning for gesture detection and localization. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8925, pp. 474–490. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16178-5_33
24. Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Vision-based fall detection with convolutional neural networks. Wirel. Commun. Mob. Comput. **2017** (2017)
25. Piccardi, M.: Background subtraction techniques: a review. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3099–3104. IEEE (2004)
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
27. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Robust video surveillance for fall detection based on human shape deformation. IEEE Trans. Circ. Syst. Video Technol. **21**(5), 611–622 (2011)
28. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1417–1426 (2017)
29. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. arXiv preprint arXiv:1704.02685 (2017)
30. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
31. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Singh, B., Marks, T.K., Jones, M., Tuzel, O., Shao, M.: A multi-stream bi-directional recurrent neural network for fine-grained action detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1961–1970. IEEE (2016)
34. Solbach, M.D., Tsotsos, J.K.: Vision-based fallen person detection for the elderly. arXiv preprint arXiv:1707.07608 (2017)
35. Stone, E.E., Skubic, M.: Fall detection in homes of older adults using the microsoft kinect. IEEE J. Biomed. Health Inf. **19**(1), 290–301 (2015)
36. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. arXiv preprint arXiv:1703.01365 (2017)

37. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3539–3548 (2017)
38. Töreyin, B.U., Dedeoğlu, Y., Çetin, A.E.: HMM based falling person detection using both audio and video. In: Sebe, N., Lew, M., Huang, T.S. (eds.) HCI 2005. LNCS, vol. 3766, pp. 211–220. Springer, Heidelberg (2005). https://doi.org/10.1007/11573425_21
39. Vishwakarma, V., Mandal, C., Sural, S.: Automatic detection of human fall in video. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 616–623. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77046-6_76
40. Wang, S., Chen, L., Zhou, Z., Sun, X., Dong, J.: Human fall detection in surveillance video based on pcanet. Multimed. Tools Appl. **75**(19), 11603–11613 (2016)
41. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. Int. J. Telemedicine Appl. **2015**, 2 (2015)
42. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
43. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2678–2687 (2016)
44. Zhang, T., Wang, J., Xu, L., Liu, P.: Fall detection by wearable sensor and one-class SVM algorithm. In: Huang, D.S., Li, K., Irwin, G.W. (eds.) Intelligent Computing in Signal Processing and Pattern Recognition. Lecture Notes in Control and Information Sciences, vol. 345, pp. 858–863. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-37258-5_104