2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

An Edge-device Based Fast Fall Detection Using Spatio-temporal Optical Flow Model

Yuchao Yang, Hongwei Ren, Chenghao Li, Chenchen Ding and Hao Yu

***Abstract*— The elderly fall detection is one critical function in health of the elderly. A real-time fall detection for the elderly has been a significant healthcare issue. The traditional video analysis on cloud has large communication overhead. In this paper, a fast fall detection system based on the *spatio-temporal optical flow model* is proposed, which is further deeply compressed by a *structured tensorization* towards an implementation on edge devices. Firstly, an object extractor is built to extract motion objects from video clips. The spatio-temporal optical flow model is formed to estimate optical flow fields of motion objects. It can extract features from objects and their corresponding optical flow fields. Then these two features are fused to form new spatio-temporal features. Finally, the tensor-compressed model processes the fused features to determine fall detection, where the strongest optical field would indicate the fall. We conduct experiments with Multicam and URFD datasets.**

***Clinical relevance*— It demonstrates that the proposed model achieves the accuracy of 96.23% and 99.37%, respectively. Besides, it attains the inference speed of 83.3 FPS and storage reduction of $210.9\times$. Our work is further implemented on an AI acceleration core based edge device, and the runtime is reduced by $9.21\times$.This high performance system can be applied to the field of clinical monitoring in the future.**

I. Introduction

Fall detection for the elderly still remains a great challenge [1]. In the literature, there are two types of fall detection systems that are currently dominant, which are 1) traditional transducer-based and 2) AI-driven vision-based approaches. The traditional sensor-based approaches commonly employ the accelerometers and gyroscopes to detect the accelerations, velocities and angles of a motion [2], [3]. Recent vision-based schemes driven by deep learning have successfully achieved fall detection in surveillance cameras. Histograms of Oriented Gradients (HOG) and Local Binary Pattern (LBP) [4] are adopted to extract hand-crafted features for fall detection. CNN-3B3Conv [5] is presented to learn features from convolutional neural networks (CNNs). These methods extract the features in the video by processing each frame, but there are some limitations in capturing temporal information along sequences. Recently, LSTM networks [6] are developed to learn temporal video representations for fall detection. 3D CNNs [7], [8] are proposed to locate the falling activities and improve accuracy of the detection by taking the place for the 2-dimensional convolutions with spatial 3-dimensional convolutions. However, all of the above methods require dense calculations of raw data, resulting in a huge amount of memory and storage needs, and hence cannot be real time on edge devices. The reluctant solution of video analysis on cloud however causes large communication overhead.

To tackle these problems, we propose a spatio-temporal optical flow model to achieve accurate and real time fall detection on edge devices. The contributions are fourfold:

- We implement the proposed fall detection network on an AI acceleration core based edge device with a real time speed.
- We devise a *spatio-temporal optical flow model* to learn the spatial features of each frame and the temporal features of optical flow field in video stream, which are further fused to the highly structured spatio-temporal features.
- We develop a *tensor-compressed LSTM* to quickly detect falling activities based on the fused spatio-temporal features.

In the following parts, the whole design of proposed fall detection network is presented in Section II. Section III provides the hardware implementation on the edge devices. Section IV reports experimental results on several different datasets and Section V summarizes the whole paper gets the conclusion.

II. Proposed Fall Detection Framework

This section presents the proposed fall detection framework, which is shown in Fig. 1. The object extractor aggregates the target objects from each frame of the surveillance videos. The spatio-temporal optical flow model learns and fuses the spatio-temporal features of the target objects in video stream. These two things together make up a Spatio-temporal feature extractor. Based on the highly structured spatio-temporal features, a tensor-compressed LSTM is further applied to precisely detect the fall.

A. Spatio-temporal Feature Extractor

The proposed object extractor divides each input frame into an $S \times S$ grid with B bounding boxes predicted with the corresponding confidence scores and each representing an object. According to the threshold value, the objects with low possibilities out of all predicted $S \times S \times B$ bounding boxes will be removed.

*This work was supported by the National Natural Science Foundation of China (NSFC) (Key Program Grant No. 62034007), Innovative Team Program of Education Department of Guangdong Province (Grant No. 2018KCXTD028), the Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B010142001) and Shenzhen Science and Technology Program (Grant No. KQTD20200820113051096).

The authors are with School of Microelectronics, and Engineering Research Center of Integrated Circuits for Next-Generation Communications, Ministry of Education, Southern University of Science and Technology, China.

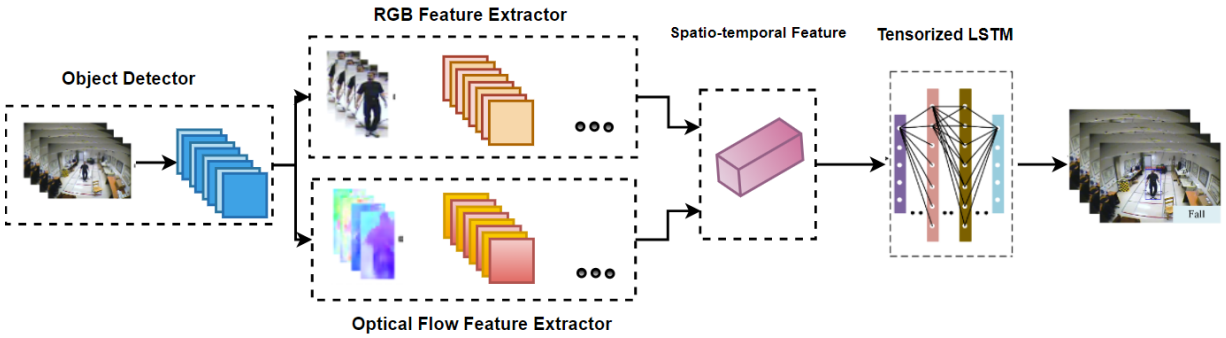

Fig. 1. The framework of fast fall detection using compressed LSTM optical flow model.

In order to make the proposed fall detection network more robust when it inferences, the optical flow features are adopted as the temporal representations to perform fall detection rather than traditional raw RGB data.

a) Optical Flow Estimation: Optical flow field is the displacement vector between continuous frames, which can represent the motion of objects. For the $k-th$ frame, $d_k(m,n)$ represents the corresponding displacement vector at the point (m,n). d_k^x and d_k^y denote the horizontal and vertical components of the vector field, respectively. The motion pattern of the objects in S continuous frames is described by stacking d_k^x and d_k^y to $2S$ channels. Specifically, the process can be indicated as follows:

$$\begin{aligned} C(m,n,2a-1) &= d_a^x(m,n) \\ C(m,n,2a) &= d_a^y(m,n) \end{aligned} \tag{1}$$

where $C \in R^{w\times h\times 2S}$ is the output of our optical flow estimation, w and h are the number of horizontal pixels and vertical pixels of the frames.

b) Spatio-temporal Features: Similar with the optical flow field, we stack the raw rgb frames to $B \in R^{w\times h\times 3S}$. The proposed spatio-temporal optical flow model can extract both spatial features and temporal features from the inputs B and C, respectively. The extraction process can be shown as follows:

$$\begin{aligned} f^{sp} &= extract_{nn}(B) \\ f^{tp} &= extract_{nn}(C) \end{aligned} \tag{2}$$

where the $extract_{nn}$ expression describes the spatial features and temporal features extraction process, f^{sp} and f^{tp} denote the extracted spatial and temporal features respectively. Furthermore, for the sake of the higher accuracy of fall detection, we fuse the f^{sp} and f^{tp} as follows:

$$f^{fused} = fuse(f^{sp}, f^{tp}) \tag{3}$$

where f^{fused} represents the fused spatio-temporal features, and the $fuse$ operation denotes the feature fusion process.

B. Tensor-compressed LSTM Model

a) LSTM Model: Based on the fused features f_t^{fused}, a tensorized LSTM is constructed to learn the distilled spatio-temporal feature for fall detection. For each cell in the tensorized LSTM, it learns to update its state parameters in real time based on the current input features and past states, as shown below:

$$\begin{aligned} I_t &= \sigma(W_i f_t^{fused} + U_i H_{t-1} + B_i) \\ D_t &= \sigma(W_d f_t^{fused} + U_d H_{t-1} + B_d) \\ E_t &= \sigma(W_e f_t^{fused} + U_e H_{t-1} + B_e) \\ \tilde{C}_t &= tanh(W_c f_t^{fused} + U_c H_{t-1} + B_c) \\ C_t &= I_t \odot C_{t-1} + D_t \odot \tilde{C}_t \\ H_t &= E_t \odot tanh(C_t) \end{aligned} \tag{4}$$

where σ indicates the sigmoid function, $\odot$ represents the element-wise product, and $tanh$ denotes the hyperbolic tangent function. H_{t-1} and C_{t-1} are the previous hidden state and previous update factor, H_t and C_t are the current hidden state and current update factor, respectively. The weight matrices W_* and U_* perform the input f_t and the previous hidden state H_{t-1} to update factor $\tilde{C}_t$ and three sigmoid gates, I_t, D_t and E_t.

The LSTM captures all each frame of video information from the initial frame till the current frame. However, the LSTM needs from huge number of parameters and high dimensional inputs, which makes it hard to train and susceptible to overfitting even when it fully converges. To address the problem, we make use of tensor decomposition method to deeply compress the LSTM.

b) Tensor Decomposition in LSTM: For a d-dimentional tensor $\mathcal{F} \in R^{l_1\times l_2\times\ldots\times l_d}$, where $\mathcal{F}(h_1,h_2,...,h_d)$ is an element specified by the indices $h_1,h_2,...,h_d$, it can be approximated by a number of tensor cores $\mathcal{G}_k \in R^{r_{k-1}\times l_k\times r_k}(k\in[1,d])$, expressed as follows:

$$\mathcal{F}(h_1,h_2,...,h_d) = \mathcal{G}_1(h_1)\mathcal{G}_2(h_2)...\mathcal{G}_d(h_d) \tag{5}$$

where $\mathcal{G}_k(h_k) \in R^{r_{k-1}\times r_k}$ is a matrix slice from the 3-dimensional tensor $\mathcal{G}_k$ and r_k is the rank of $\mathcal{G}_k$. For each integer l_k, it can be further decomposed as $l_k = m_k \times n_k$. As a result, each tensor core $\mathcal{G}_k$ can be transformed and reformed into $\mathcal{G}_k^* \in R^{r_{k-1}\times r_k\times m_k\times n_k}$. Therefore, the decomposition of the d-dimentional tensor $\mathcal{F} \in R^{m_1\times n_1\times m_2\times n_2\times\ldots m_d\times n_d}$ can be rewritten as:

$$\begin{aligned} &\mathcal{F}((i_1,j_1),(i_2,j_2),...,(i_d,j_d)) \\ &= \mathcal{G}_1^*(i_1,j_1)\mathcal{G}_2^*(i_2,j_2)...\mathcal{G}_d^*(i_d,j_d) \end{aligned} \tag{6}$$

Double-index trick [9] as such is the core function to decompose the fully-connected computation in LSTM cells.

In general, the large-scale matrix-vector multiplication is the most expensive computation in the LSTM cell, generically denoted as:

$$y = Wx + b \tag{7}$$

where $x \in R^N$ is the input vector, $b \in R^M$ is the bias vector, $y \in R^M$ is the output vector, and $W \in R^{M \times N}$ is the weight matrix. To approximates Wx with much fewer parameters, the weight matrix W is reshaped into a tensor $\mathcal{W} \in R^{(m_1 \times m_2 \times ... \times m_d) \times (n_1 \times n_2 \times ... \times n_d)}$, where $M = \prod_{k=1}^{d} m_k$ and $N = \prod_{k=1}^{d} n_k$. Similarly, x and b can be reshaped into d-dimentional tensors $\mathcal{X} \in R^{n_1 \times n_2 \times ... \times n_d}$ and $\mathcal{B} \in R^{m_1 \times m_2 \times ... \times m_d}$. As a result, the output y can also become a d-dimentional tensor $\mathcal{Y} \in R^{m_1 \times m_2 \times ... \times m_d}$. Therefore, the matrix-vector multiplication in LSTM cells can be reformulated as:

$$\mathcal{Y}(i_1, i_2, ..., i_d) = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} ... \sum_{j_d=1}^{n_d} [\mathcal{G}_1^*(i_1, j_1) \mathcal{G}_2^*(i_2, j_2) ... \mathcal{G}_d^*(i_d, j_d) \mathcal{X}(j_1, j_2, ..., j_d)] + \mathcal{B}(i_1, i_2, ..., i_d) \tag{8}$$

Due to the tensor decomposition strategy, the computational complexity in the tensor-compressed LSTM turns out to be $O(dr_{max}^2 m_n)$ instead of $O(m_n^d)$, where r_{max} is the maximum rank of cores $\mathcal{G}_k$ and m_n is the maximum model size $m_k \cdot n_k$ of weights $\mathcal{W}$.

III. Implementation on Edge Device

To verify the speedup and lightweight of our proposed fall detection network, we implemented it on an edge device with the high efficient and programmable AI chip. As shown in Fig. 2, the chip contains a digital vision pre-processing module, an Arm CPU, two AI cores, an on-chip buffer with the capacity of 8MB, and some driver interfaces. It delivers up to a peak performance of about 8 TFLOPs for 16-bit floating-point and 16 TOPs for quantized 8-bit computations, with merely $8W$ of power consumption. The AI core is specifically deployed as a neural network accelerator.

Due to complicate network architecture, the feature extraction part and the LSTM prediction part are the most time-consuming parts in our fall detection network. To improve their runtime efficiency on the AI acceleration core based edge device, we choose two different accelerating methods. AI acceleration core plays an important role in speeding up the feature extraction part. As for the LSTM prediction part, tensor compression is utilized to accelerate it. The experimental results are presented in detail in section V.

IV. Experiments

The experimental setups are following: 1) NVIDIA GTX-1080Ti, CPU E5-2660 and AI acceleration core are employed for hardware realization and performance comparison. 2) The tensor rank r_k is set to 4. 3) The training epochs are set to 300 on the Multiple Cameras Fall Dataset (Multicam) while 200 on the UR Fall Dataset (URFD).

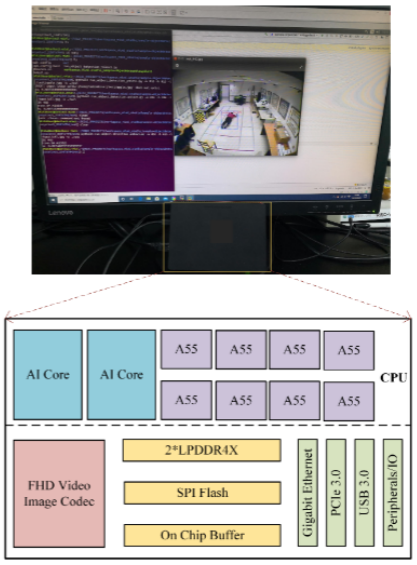

Fig. 2. Running compressed LSTM optical flow model on edge device platform.

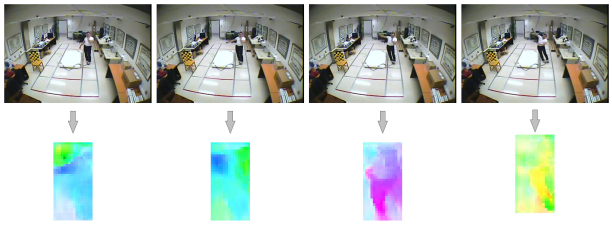

Fig. 3. Continuous frame samples and corresponding optical flow.

A. Performance Analysis

Multicam is a widely used fall detection dataset in the literature. It contains 24 scenarios recorded with 8 video cameras in 8 different locations. The accuracy comparison between our method and some state-of-the-art approaches on the Multicam is shown in Table I. It can be observed in Table I that our method maintains the classification accuracy as 96.23%, which outperforms all the state-of-the-art approaches. It is 1.92% higher than the second best one. And the accuracy has improved by 2.22% compared with the Silhouette Area Variation [10] method which is based on a layered hidden Markov model (LHMM). Specifically, an increment of 13.07% against the deep-learning-based approach Simple CNN [11].

URFD is also an often-used dataset for fall detection. It contains 70 sequences. Table II shows the accuracy results of the proposed method and the state-of-the-art approaches on URFD. It can be seen that the proposed method significantly outperforms others. Specifically, the accuracy of our method reaches 99.37%, which is 3.17% higher than Hourglass Convolution [12] and 4.37% than the MEWMA [13].

Fig. 3 shows visual samples of the outputs. As a result, we can obtain abundant temporal information from the optical flow field outputs. The sample visual results on the Multicam are shown in Fig. 6. One can observe that all objects can be detected and all "fall" or "no fall" activities can be precisely

TABLE I

ACCURACY COMPARISON ON THE MULTICAM

Method	Sensitivity	Specificity	Accuracy
Bounding box ratio [14]	-	-	56.60%
Simple CNN [11]	71.67%	77.48%	82.26%
Skeleton feature [15]	90.9%	93.75%	92.59%
Silhouette Area Variation [10]	-	-	94.01%
PA_Ada [16]	93.85%	91.85%	94.31%
Ours	**93.10%**	**98.87%**	**96.97%**

TABLE II

ACCURACY COMPARISON ON THE URFD

Method	Sensitivity	Specificity	Accuracy
Shi-Tomasi algorithm [22]	96.66%	95.00%	93.54%
MEWMA [13]	100.00%	92.00%	95.00%
Hourglass Convolution [12]	100.00%	93.00%	96.20%
Zerrouki and Houacine [23]	-	-	96.88%
Ours	**100.00%**	**98.51%**	**99.37%**

predicted.

Besides the outstanding accuracy of our fall detection network, the storage size and inference speed are also remarkable. Fig. 4 and Fig. 5 show the storage size and speed comparisons. The storage size of model is reduced by $210.9\times$ after tensor compression. Furthermore, it also shows superiority against other advanced approaches shown in the Fig. 4, such as TSM [17], ST-ResNet [18] and 3D-CNN based approaches [8]. From Fig. 5, we observe that the proposed network can run at 83.3 fps, which is faster than all other approaches, including iDT+FV [19], Brox's Flow [20], BoCSS+VPSO-ELM [21] and 3D-CNN based approaches [7]. Since the storage size is significantly reduced and the inference speed is highly accelerated, the proposed fall detection network is suitable for implementation on edge devices.

B. Evaluation on Edge Device

We further report the performance of the proposed fall detection network on an AI acceleration core based edge device. A 720×480 video is used to test the run time of inference. As shown in Table III, benefiting from the AI acceleration core and tensor compression method, the proposed framework achieves $9.21\times$ speedup compared with a high-end CPU.

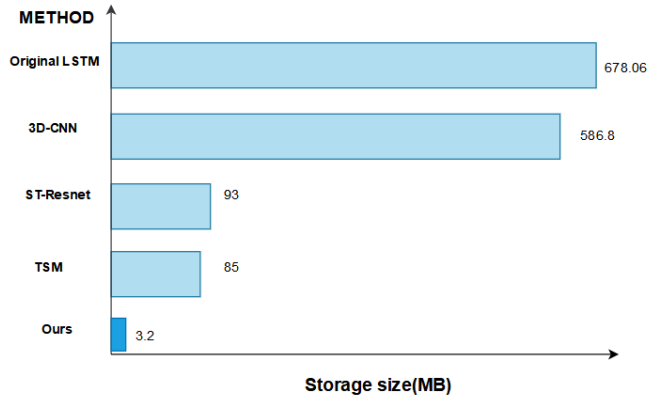

Fig. 4. Storage size comparison.

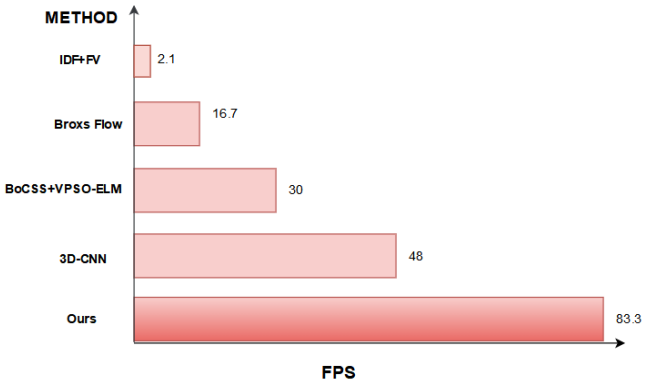

Fig. 5. Inference speed comparison.

TABLE III

RUN TIME COMPARISON OF OUR MODEL ON AI CORE BASED EDGE DEVICE

	Method	Run time
CPU	Feature extraction + LSTM	30.53s
Arm CPU + AI core	Feature extraction + T-LSTM	2.99s

V. CONCLUSIONS

This paper has proposed a real time edge device based fall detection network using sptio-temporal optical flow model which extracts the objects in the raw rgb frames. Then a tensor-compressed LSTM is built based on the spatio-temporal features to detect "fall" activities. The two techniques can significantly improve the accuracy, inference speed and compression ratio of our fall detection network. Using the benchmarks of Multicam and URFD, the proposed fall detection network achieves the accuracy of 96.23% and 99.37%, respectively. Moreover, the inference speed reaches 83.3 fps with $210.9\times$ reduction in storage size. According to superior performance, it is realized on an AI core based edge device. The whole system achieves $9.21\times$ speedup compared with a high-end CPU, thereby rendering it a strong candidate for fall detection on edge devices.

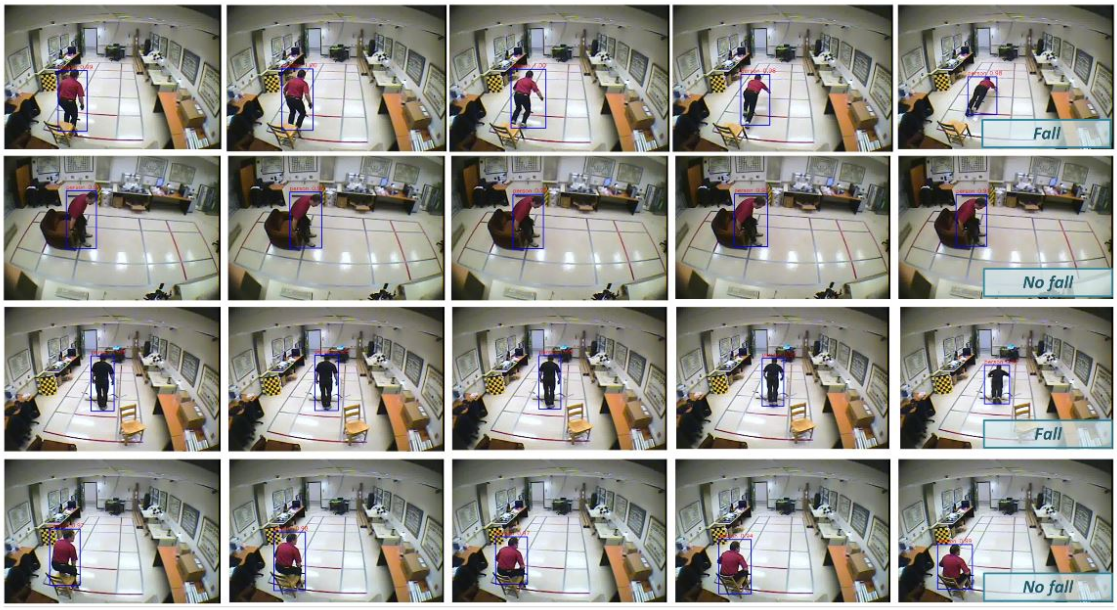

Fig. 6. Video sample of fall detection results on edge device.

REFERENCES

[1] S. Brownsell and M. S. Hawley, "Automatic fall detectors and the fear of falling," *Journal of telemedicine and telecare*, vol. 10, no. 5, pp. 262–266, 2004.

[2] P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa, "Evaluation of fall detection for the elderly on a variety of subject groups," in *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology*, 2009, pp. 1–4.

[3] M. Luštrek, H. Gjoreski, S. Kozina, B. Cvetković, V. Mirchevska, and M. Gams, "Detecting falls with location sensors and accelerometers," in *Twenty-Third IAAI Conference*, 2011.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[5] G. L. Santos, P. T. Endo, K. H. d. C. Monteiro, E. d. S. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, p. 1644, 2019.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] N. Lu, X. Ren, J. Song, and Y. Wu, "Visual guided deep learning scheme for fall detection," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2017, pp. 801–806.

[8] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Multi-fiber networks for video recognition," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 352–367.

[9] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Advances in neural information processing systems*, 2015, pp. 442–450.

[10] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A lhmm-based approach," *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 11, pp. 1522–1532, 2008.

[11] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, and E. Moya-Albor, "Application of convolutional neural networks for fall detection using multiple cameras," in *Challenges and Trends in Multimodal Fall Detection for Healthcare*. Springer, 2020, pp. 97–120.

[12] X. Cai, S. Li, X. Liu, and G. Han, "Vision-based fall detection with multi-task hourglass convolutional auto-encoder," *IEEE Access*, vol. 8, pp. 44 493–44 502, 2020.

[13] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "Vision-based fall detection system for improving safety of elderly people," *IEEE Instrumentation & Measurement Magazine*, vol. 20, no. 6, pp. 49–55, 2017.

[14] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on circuits and systems for video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[15] Y.-T. Chen, Y.-C. Lin, and W.-H. Fang, "A hybrid human fall detection scheme," in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 3485–3488.

[16] S. F. Ali, R. Khan, A. Mahmood, M. T. Hassan, and M. Jeon, "Using temporal covariance of motion and geometric features via boosting for human fall detection," *Sensors*, vol. 18, no. 6, p. 1918, 2018.

[17] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[18] R. Christoph and F. A. Pinz, "Spatiotemporal residual networks for video action recognition," *Advances in Neural Information Processing Systems*, pp. 3468–3476, 2016.

[19] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[20] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European conference on computer vision*. Springer, 2004, pp. 25–36.

[21] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE journal of biomedical and health informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[22] S. Bhandari, N. Babar, P. Gupta, N. Shah, and S. Pujari, "A novel approach for fall detection in home environment," in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2017, pp. 1–5.

[23] N. Zerrouki and A. Houacine, "Combined curvelets and hidden markov models for human fall detection," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 6405–6424, 2018.