2018 IEEE International Conference on Big Data (Big Data)

SPATIO-TEMPORAL CONVOLUTIONAL NEURAL NETWORK FOR ELDERLY FALL DETECTION IN DEPTH VIDEO CAMERAS

Maryam Rahnemoonfar, Hend Alkittawi*

Computer Vision and Remote Sensing Laboratory (Bina Lab)
Department of Computing Sciences
Texas A&M University-Corpus Christi, USA

ABSTRACT

Emergency departments treat around 2.5 million older people for fall injuries each year. Preserving the elderlys' right of aging in a home of their own choice is mandatory in today's world, as more elderly people are willing to live independently. Current implementations of fall detection systems lack accuracy. Despite efforts to detect elderly falls, it is possible that daily life activities, such as lying down, trigger false alarms. Moreover, privacy is the main concern for visual cameras. In this research we used deep convolutional neural networks to describe the overall space-time appearance pattern of a fall-event in depth video cameras. We developed a 3D convolutional neural network to capture both the spatial information available in video frames, and the temporal information presented through successive video frames. Our method outperformed the state-of-the art accuracy with a large margin.

1. INTRODUCTION

Preserving the elderlys' right of aging in a home of their own choice is mandatory in today's world, as more elderly people are willing to live independently. But, with the statistics showing that falling is a major health problem that has a huge non-desirable impact on elderly lives, fall detection systems become a necessity.A variety of systems have been developed to detect falls [1][2]. Most of the available systems are based on image processing and computer vision algorithms. However privacy is the main concern for visual cameras. The advantage that depth cameras provide is preserving the privacy of people under surveillance, since color images are not used. The current existing techniques for fall detection based on computer vision is based on traditional feature engineering techniques. These features can be as simple as the ratio between the width and height of the bounding box surrounding a human [3], and as complicated as the distance of the points in a human point cloud to the floor [4]. Some other features include extracting the patterns of change in human curvature [5], or human silhouette orientation [6] during a fall event. Designing the best features in each case is a difficult task and need extensive knowledge of movement patterns and the environment. In recent years deep learning algorithm have been extensively used for hierarchical feature representation and learning.

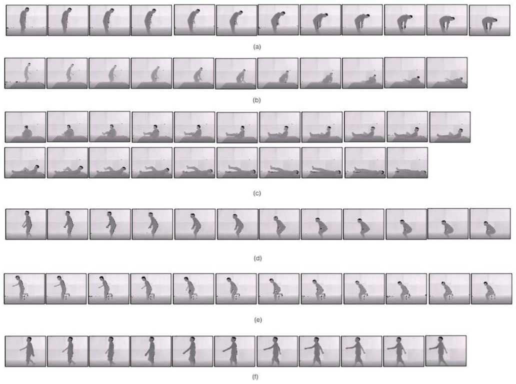

Fig. 1. Sample video frames. (a) Bending frames. (b) Falling frames. (c) Lying frames. (d) Squatting frames. (e) Sitting frames. (f) Walking frames.

Convolutional neural networks are extensively used for classification [7, 8, 9, 10], object recognition [11, 12], counting [13, 14], contour and edge detection [15] and semantic segmentation [16, 17, 18]. Due to the magnificent performance of these algorithms, human action recognition has received extensive interest by deep learning researchers. More recently, deep learning techniques have been used to recognize actions in RGB videos [19]. These applications are flourishing with the increased processing capabilities available through GPUs, as well as the large video datasets that were made available for the research community.

In this research we have used a depth action dataset collected with Kinect camera. A sample of frames for the different actions is shown in Figure 1. For detecting fall events in depth video cameras, we designed a novel spatial-temporal convolutional neural network that is able to detect fall actions with high accuracy in depth video cameras. We outperformed state-of-the art accuracy with a large margin.

*Corresponding author.

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

2. RELATED WORKS

Fall detection systems can be generally categorized under the following types: context-aware systems, wearable devices, and cellphone based systems [1]. In context-aware systems, sensors are deployed in the environment to detect falls. The most commonly used sensors, in more recent research, are wide angle cameras[20], and depth cameras [4][21][6][5]. Installing cameras with other sensors such as microphones [22] and pressure sensors [23] is also a common approach. Radars are another type of sensor that is used in context-aware systems to sense motions based on the wavelet transform [24]. Wearable devices can be defined as miniature electronic sensor-based devices that are worn by the bearer, under, with, or on top of clothing. For these systems accelerometers are mainly used [25]. Approaches that uses cellphones for fall-detection take advantage of the accelerometers embedded in mobile devices. These approaches also use the built-in communication functionality in cellphones for designing fall-detection systems [26].

Context-aware systems in general are superior over wearable devices in terms of reliability as there are no chances of forgetting or losing them. Occlusion represents a considerable constraint for systems based on computer vision [21]. The advantage that depth cameras provide is preserving the privacy of people under surveillance, since color images are not used. To classify the events as falls or non-falls, techniques such as Decision Trees [21], Support Vector Machine [27], Kalman Filtering [20], Thresholding Techniques [4], and Nearest-neighbor Rule [4] are used. Overall, the selection of a classification technique often depends on the set of features obtained to map an input to an output. For fall-detection systems that are based on computer vision approaches, these features can be as simple as the ratio between the width and height of the bounding box surrounding a human [3], and as complicated as the distance of the points in a human point cloud to the floor [4]. Some other features include extracting the patterns of change in human curvature [5], or human silhouette orientation [6] during a fall event. The focus of the research in this area is to design the best features that would correctly identify fall events, with the minimum false alarms ratio.

Deep learning algorithms allow the system to learn the best features to recognize human actions in videos. One class for human action recognition using deep learning relies on applying convolutional neural networks to process video frames [19][28][29]. The work presented in [19] uses Sports-1M dataset with one million videos representing 487 human action classes. In their work, the authors stack the video frames and feed them to a multi-resolution CNN. The multi-resolution network has two streams for processing; a low-resolution stream and a high-resolution stream. The authors also have investigated multiple approaches for fusing temporal data in the convolutional neural network. The approaches include a single-frame model, early-fusion, late-fusion and slow-fusion. In [28], the authors use two-stream convolutional neural network. One stream is used to capture the spatial information by processing single frames. The other stream is used to capture the temporal information by processing multi-frame optical flow representations. A 3D convolutional neural network was used in [29] to extract the spatial and temporal information encoded in successive video frames. The datasets used to test this 3D architecture are the TRECVID 2008 and the KTH with more than 49-hour videos. Bounding boxes were drawn around humans in the scene to keep track of them. Video frames were fed into the network which consists of two 3D convolutional layers, two pooling layers, one 2D convolutional layer, and one fully connected layer. In most cases, the classification based on the CNN approach achieves good results. For RGB video frames, these results can be due to the association of human actions with the presence of certain objects in the scene. For example, for a human action to be classified as swimming, a water surface needs to be present in the scene.

3. METHODOLOGY

Convolutional Neural Networks, (CNNs), are a subclass of feedforward neural networks that are mainly used for processing images. The building block of a CNN is the convolutional layer. This layer has neurons connected to local regions in the input image and it is where most of the computations are performed. The convolutional layer represents filters whose parameters are learned by the network. The pooling layer of the CNN is used to down sample the activation volume that results from the convolutional layer. This has an important role for reducing the amount of computations. For image classification problems, the fully connected layer is used to compute the scores of classes. The neurons of this layer are connected to all activations in the previous layer. The output of this layer is a vector holding the class prediction for an input image. To turn these predictions into class probabilities, a softmax function is applied.

In this study we designed a spatial-temporal 3D convolutional neural network for fall-detection. The input to the network is a video of size $99 \times 160 \times 120 \times 3$. The video is then goes through the first 3D convolutional layer. This layer has 96 filters each of size $25 \times 11 \times 11$, where the filter spans the videos with a stride of 3 in the spatial domain and a stride of 3 in the temporal domain. The size of the filter in the temporal domain was chosen such that the network looks at sufficient number of frames before updating it's parameters. So, based on the conducted experiments using around one-fourth the number of video frames served this purpose. Next, we have a max-pooling layer that down-samples the resulting activation volume by 2 in every dimension. Following this pooling layer we have a second convolutional layer of 256 filters each of size $15 \times 5 \times 5$ that span the video with a stride of 2 in both spatial and temporal domains. Again, a pooling layer, with the same properties as the first pooling layer, follows this convolutional layer. We then have three convolutional layers of sizes $5 \times 3 \times 3$, $1 \times 3 \times 3$ and $1 \times 3 \times 3$ respectively. These convolutional layers are followed by a pooling layer. A summary of the filter sizes and strides for each layer is listed in table 1. Finally, a fully-connected layer of 128 neurons is followed by another fully connected layer whose number of neurons is associated with the number of classes in the problem are used. A softmax function is applied to the output of the last fully-connected layer to compute the classes' probabilities.

Layer	Spatial Filter		Temporal Filter	
	Size	Stride	Size	Stride
Conv1	11	3	25	3
Pool1	2	2	2	2
Conv2	5	2	15	2
Pool2	2	2	2	2
Conv3	3	1	5	3
Conv4	3	1	1	2
Conv5	3	1	1	2
Pool3	2	2	1	2

Table 1. A summary of the filter sizes and strides for each layer in the deep fall-detection system.

To train the network, Adam Optimizer algorithm was used. This is a gradient-based algorithm which requires little memory and is very efficient in terms of computations. The use of Adam optimizer fits problems of large data and parameters. Practically, this optimizer works well compared to other stochastic optimization methods [30]. The loss function used is the cross-entropy function given by equation 1. Using this function the optimizer minimizes the sum of the difference between labels and predictions of all samples in a batch.

$$Cost \ = - \sum_{i=2}^{i=N} Label_i - log(Prediction_i) \qquad (1)$$

After experimenting with different hyper-parameters, the hyper-parameters of the network were set as follows: 0.7 for the dropout rate, 10 for the batch size, and 1e-4 for the learning rate. Two values for the dropout rate were evaluated; 0.5 and 0.7. Using the dropout technique means that the network randomly and temporarily removes some of the network units and their connections. Thus, the higher dropout rate is, the lower overfitting the network has [31]. Two values for the learning rate were also used for performance comparison, 1e-4 and 1e-8. The learning rate is used by the optimizer to adjust the weights and biases of the network. Different combinations of these values were tested, with accuracies below 70% for all combinations other than 0.7 for dropout rate, and 1e-4 for the learning rate. Several values for the batch size were tested.

The values of the batch size with a training set of 810 videos were: 10, 15, 18, and 30. The values of the batch size for a testing set of 265 videos was set to 5.

4. EXPERIMENTAL RESULTS

This study uses the SDUFall dataset[5] which contains 6 actions performed by twenty young people. These action are: bending, falling down, lying, squatting, sitting and walking. Men and women participants performed each action multiple times. The conditions under which these actions were captured are different based on the lighting conditions, and the direction and position of the stunt relative to the camera. The Microsoft Kinect sensor was installed at 1.5m height. The videos were recorded at 30 frames per second, with a 640×480 frame size. On average, the length of a video is 5.6 seconds.

The results for depth videos classification were obtained based on two methods. One method distinguishes between all six classes presented in the dataset. The other classifies fall events among non-fall events. Following are the results in details.

4.1. Six-Class Classification

In this method, each video was given a label as bending, falling, lying-down, sitting, squatting, or walking. For the training phase, 135 videos were used per each action. The overall training time was around 32 minutes. A total of 265 videos were used for testing. The number of videos used for testing per class is as follows: 34 videos for bending, 55 videos for falling, 35 videos for lying-down, 57 videos for sitting, 46 videos for squatting, and 38 videos for walking.

The estimated time for classifying one video is around 23 seconds. So, the system could achieve real time performance. The confusion matrix for the 6-class action recognition is shown in Table 2.

The system could achieve 87.28% accuracy on fall events classification. The system could also achieve high specificity by correctly identifying daily life activities as non-falls. Interestingly, all lying-down events were correctly classified. The majority of miss classified falling events are given a lying-down label. This can be due to the fact that the dataset have simulated falls. Therefore, some of the fall events were performed more like lying-down events rather than falling events.

Samples of the correctly classified and miss classified falling video frames are shown in Figures 2 and 3, respectively.

It can be seen that the miss-classified video frames share some patterns with a lying-down action, where the human

93.2%	Bending	Falling	Lying-Down	Sitting	Squatting	Walking
Bending	91.18	0	0	0	8.82	0
Falling	0	87.28	12.72	0	0	0
Lying-Down	0	0	100	0	0	0
Sitting	3.51	0	0	94.74	0	1.75
Squatting	8.70	0	0	0	91.30	0
Walking	2.63	0	0	0	0	97.37

Table 2. The confusion matrix for the 6-class action recognition. It shows the percentages for accuracies.

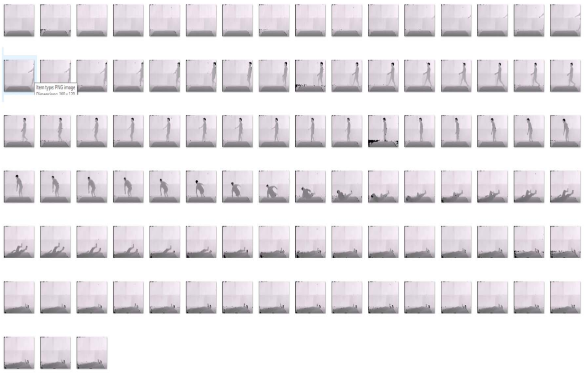

Fig. 2. Sample of a correctly classified falling video frames using the six-class classification method.

bends towards a mattress.

4.2. Fall versus Non-Fall Classification

The goal of this research is to correctly classify fall-events as fall-events, otherwise, events should be classified as non-fall events. Using this approach, each video was given a label, either fall or non-fall. The average accuracy of the system is 97.58%. To the best of our knowledge, this is the highest accuracy reported for fall-event recognition on SDUFall dataset.

Table 3, shows a comparison of our approach to approaches which uses hand-crafted features such as the orientation volume and the curvature scale space of human silhouettes.

5. CONCLUSION

In conclusion, the research presented in this study shows that deep learning based algorithms are suitable for recognizing fall-event patterns. It is observed that the deep learning based pattern representations help increasing the accuracy for fall-detection systems as compared to traditional pattern representation techniques such as the orientation volume and the curvature scale space of human silhouettes. The main idea in this research is to identify the aforementioned patterns using spatio-temporal features. These spatio-temporal features were automatically learned and extracted using 3D convolution neural network.

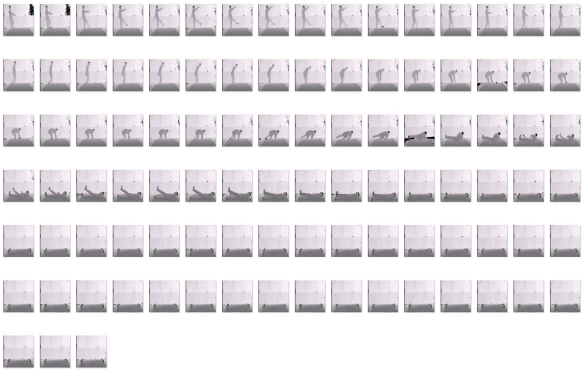

Fig. 3. Sample of a miss- classified falling video frames using the six-class classification method.

Methods	Accuracy
BoW-VPSO- ELM[5]	86.83
FV-SVM[32]	88.83
BoSOV-Bayes[6]	91.89
Our method	**97.58**

Table 3. Comparison of the performance of the fall vs. non-fall action recognition. It shows the percentages for accuracies.

6. REFERENCES

[1] Raul Igual, Carlos Medrano, and Inmaculada Plaza, “Challenges, issues and trends in fall detection systems,” *Biomedical engineering online*, vol. 12, no. 1, pp. 1, 2013.

[2] Muhammad Mubashir, Ling Shao, and Luke Seed, “A survey on fall detection: Principles and approaches,” *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[3] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline 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.

[4] Bogdan Kwolek and Michal Kepski, “Improving fall detection by the use of depth sensor and accelerometer,” *Neurocomputing*, vol. 168, pp. 637–645, 2015.

[5] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji, and Yibin Li, “Depth-based human fall detec-

tion via shape features and improved extreme learning machine," *IEEE journal of biomedical and health informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[6] Erdem Akagunduz, Muzaffer Aslan, Abdulkadir Sengur, Haibo Wang, and Melih Ince, "Silhouette orientation volumes for efficient fall detection in depth videos," *IEEE journal of biomedical and health informatics*, 2016.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.

[8] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[9] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[10] Clay Sheppard and Maryam Rahnemoonfar, "Real-time scene understanding for uav imagery based on deep convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2017, pp. 2243–2246.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[12] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik, *Simultaneous Detection and Segmentation*, pp. 297–312, Springer International Publishing, Cham, 2014.

[13] Maryam Rahnemoonfar and Clay Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, pp. 905, 2017.

[14] Maryam Rahnemoonfar and Clay Sheppard, "Real-time yield estimation based on deep learning," in *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II*. International Society for Optics and Photonics, 2017, vol. 10218, p. 1021809.

[15] Hamid Kamangir, Maryam Rahnemoonfar, Dugan Dobbs, John Paden, and Geoffrey C Fox, "Detecting ice layers in radar images with deep hybrid networks," in *Proceedings of the IEEE Conference on Geoscience and Remote Sensing (IGARSS)*, 2018.

[16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[17] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3376–3385.

[18] Maryam Rahnemoonfar, Murphy Robin, Marina Vicens Miguel, Dugan Dobbs, and Ashton Adams, "Flooded area detection from uav images based on densely connected recurrent neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*. IEEE, 2017, pp. 3743–3746.

[19] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[20] Khosro Rezaee, Javad Haddadnia, and Ahmad Delbari, "Modeling abnormal walking of the elderly to predict risk of the falls using kalman filter and motion estimation approach," *Computers & Electrical Engineering*, vol. 46, pp. 471–486, 2015.

[21] Erik E Stone and Marjorie Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE journal of biomedical and health informatics*, vol. 19, no. 1, pp. 290–301, 2015.

[22] Yun Li, KC Ho, and Mihail Popescu, "A microphone array system for automatic fall detection," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 5, pp. 1291–1301, 2012.

[23] Huan-Wen Tzeng, Mei-Yung Chen, and Jai-Yu Chen, "Design of fall detection system with floor pressure and infrared image," in *2010 International Conference on System Science and Engineering*. IEEE, 2010, pp. 131–135.

[24] Bo Yu Su, KC Ho, Marilyn J Rantz, and Marjorie Skubic, "Doppler radar fall activity detection using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 3, pp. 865–875, 2015.

[25] Lina Tong, Quanjun Song, Yunjian Ge, and Ming Liu, "Hmm-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1849–1856, 2013.

[26] Mohammad Ashfak Habib, Mas S Mohktar, Shahrul Bahyah Kamaruzzaman, Kheng Seang Lim, Tan Maw Pin, and Fatimah Ibrahim, "Smartphone-based solutions for fall detection and prevention: challenges and open issues," *Sensors*, vol. 14, no. 4, pp. 7181–7208, 2014.

[27] Marc Bosch-Jorge, Antonio-José Sánchez-Salmerón, Ángel Valera, and Carlos Ricolfe-Viala, "Fall detection based on the gravity vector using a wide-angle camera," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7980–7986, 2014.

[28] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[29] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[30] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[32] Muzaffer Aslan, Abdulkadir Sengur, Yang Xiao, Haibo Wang, M Cevdet Ince, and Xin Ma, "Shape feature encoding via fisher vector for efficient fall detection in depth-videos," *Applied Soft Computing*, vol. 37, pp. 1023–1028, 2015.