CS482/682 Final Project Report Group 19

DetectaTrip: Early Detection of Unexpected Falls

Erica Tevere, Kinjal Shah, Maia Stiber, Catalina Gomez
(etevere1, kshah31, mstiber1, cgomezc1)

1 Introduction

Our work aims to localize and detect onset of falls from the video. We expand on previous work using contrived datasets to detect falls in highly variable video. Detection of falls can support increased response times to prevent injury in elderly populations, individuals who suffer from seizures, those with limited mobility, and patients in the hospital.

2 Methods

Datasets We trained and evaluated our method on the UR Fall Detection (URFD) Dataset (1) that is comprised of daily life activity (ADL) (40) and simulated fall events (30) RGB video sequences with annotations indicating fall onset and termination. Average fall length was $1.97s$, and the annotated falls corresponded to 15% of frames, which demonstrated the data imbalance.

A limitation of current methods is the absence of realistic fall datasets. *Oops!* dataset (2) allows us to study falls in realistic settings. We use the labeled onset of unintended action and captions to extract subsets of falls and ADL.

Approach We investigated Convolutional LSTM Autoencoder (CLA) and 3D-CNNs due to the importance of temporal information. CLA was previously used on fall datasets including URFD (3). The pretrained model was trained on non-fall, and tested on fall and non-fall depth video. During testing, the MSE between the predicted and observed next frame was used to determine fall onset and termination.

We evaluated 3D-CNNs with spatio-temporal kernels. 3D-CNNs are harder to train due to the additional kernel dimension increasing the parameter count. Thus, we built on an action recognition network by Hara *et al.* (4), pretrained on the Kinetics-700 dataset (5), and fine-tuned a 3D ResNet-18 for fall detection. We first defined the problem as a frame-wise binary classification, with input being a sequence of a frames and the output is a fall/no fall class score for the central frame. We also proposed a frame-wise voting setup, in which the model outputs a fall class score for each input frame. Supervision in both cases was provided by the frame annotations.

Implementation Details For training the 3D CNN, we used SGD with an initial learning rate of $1e-3$, an L_2 coefficient of $1e-3$, momentum of 0.9, batch size of 6 (12 for *Oops!*) and cross-entropy loss. We augmented data with flips and random crops, and resized spatial dimensions to match the pretrained model input size. To address class imbalance, we defined a sampling strategy for the fall videos where the central frame of the input sequence was uniformly sampled from fall frames. The central frame for no fall examples was randomly sampled from ADL. We generated 16-frame windows around the central frame, padding with empty frames when needed. To augment the training data, we sampled multiple examples from each video distributed across batches, and added a perturbation to onset frames to provide diversity within each iteration.

During inference, to generate predictions for complete videos, we used a 16-frame sliding window with a stride of 1, so that each frame was assigned a class score, which was then changed into a binary prediction with a SoftMax.

Evaluation For CLA, we used ROC AUC as the metric. We used precision, recall, and F1-score to

evaluate the 3D CNN's performance on fall detection, since these are robust to class imbalance. We also computed the absolute error between predicted onset fall frame and ground truth.

3 Results

ConvLSTM We compares the CLA models trained on different datasets (URFD-Depth, SDU, and Thermal) using ROC AUC which indicated the ability to detect falls (3). In our results, we achieved similar performance on RGB data as the depth input used in the paper (see Table 1). Evaluation of a subset of falls from the *Oops!* Dataset reveals comparatively minor decline in performance from URFD-RGB given drastically different inputs. Our best model has an average fall onset error of 22.4 frames (0.75s) on URFD.

3D-ResNet-18 Table 1 shows results of our validation set. The best model (URFD) achieved a F1-score of 0.92 and average fall onset error of 8.7 frames (0.29s) on test set. We validated extension of models trained for action recognition on the Kinetics dataset to fall onset detection, by adapting the network to produce a frame-based classification. As this task served as a proof-of-concept for our hypothesis, we attempted to generalize this method to the *Oops!* dataset. However, on training using 200 videos (50-50 fall-no fall, 80-10-10 train-val-test), we achieved near random performance (val. accuracy of 50%). However, we attribute this to quality of annotations provided with the dataset.

Table 1: Overall results of the CLA (ROC AUC value shown) and 3D CNNs.

Models	URFD-D	URFD-RGB	OOPS!
CLA (Thermal)	0.66	**0.59**	0.33
CLA (URFD-D)	**0.81**	0.51	**0.45**
CLA (SDU)	0.73	0.53	0.42
3D ResNet-18 URFD			
Experiment	F1	Precision	Recall
1 output-FC	0.92	0.92	0.93
Frame voting-FC	**0.95**	0.92	0.98
Frame voting-DO	0.91	0.87	0.97
3D ResNet-18 Oops!			
1 output-FC	0.40	0.34	0.68

4 Discussion

Through the analysis presented in this work, we successfully were able to localize falls from video, as well as determine the fall onset on the URFD datasets. Our results on the *Oops!* dataset show promising results, and we expect that by improving data quality (correctly clipping frames, self annotating falls, etc.), our performance will also improve. Our results show that pretrained networks for action recognition can be adapted for fall onset detection.

The limited size of the URFD dataset resulted in overfitting of the 3D CNN, which we tried to alleviate by adding regularization to the model and stochasticity to the data sampling. To preserve the robust representations of a pretrained model on a large-scale video dataset, we froze the generic features while retraining the last layer (FC) for our task. The model still achieved accurate detections on unseen data, which we attributed to the constrained conditions and similarity of the data.

This work highlights the impact of datases on fall detection performance. Current methods (3) use simulated datasets, where true conditions are not mode. *Oops!* in-the-wild videos are more challenging, and there is a wide variation in the quality of the action onset and termination label, and captioning.

References

[1] B. Kwolek *et al.* [Online]. Available: http://fenix.univ.rzeszow.pl/ mkepski/ds/uf.html

[2] D. Epstein *et al.* [Online]. Available: https://oops.cs.columbia.edu/data/

[3] J. Nogas *et al.*, "Deepfall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders," *Journal of Healthcare Informatics Research*, 2020.

[4] K. Hara *et al.*, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *CVPR*, 2018.

[5] L. Smaira *et al.*, "A short note on the kinetics-700-2020 human action dataset," 2020.