Two-Stream ConvNet for Video-Based Fall Detection

Weixuan Gao, Jie Wu, Qianmin Hu
Stanford University
`{gaow, jiewu22, qianmin}@stanford.edu`

Abstract

The project applies a two-stream convolutional neural network to detect human fall actions in indoor surveillance videos. The first stream of the two-stream model captures the spatial patterns in input data by selecting frames from each video and feed the frames to a convolutional neural network. The second stream captures temporal information by calculating optical flows between consecutive frames. After experimenting on ResNet18, ResNet101 and ResNet152, and comparing the evaluation metrics of the two streams and the fusion result, we find that the spatial stream with ResNet152 yields the highest F1 score of 91.77%.

1 Introduction

Human activity recognition in videos is one of the most important and challenging fields of study in computer vision. Among various type of human actions, fall recognition has increasingly drawn researchers' attention, since fall is among the most dangerous situations for elderly people at home and other settings, resulting in fatal injuries and loss of independence of the elderly [1]. It is crucial to know that elders fell down as soon as possible in order to provide them with emergent medical service. Automatically detecting falls and sending alarms as soon as possible could potentially minimize the injury and save lives. Furthermore, falls detection has wide application in other circumstance such as children care and public security. Video-based fall detection method is promising considering rapid development of computer vision and high penetration rate of security camera in recent years.

We apply a two-stream model on a video binary classification problem. With videos as original input, the first stream of our model feeds frames extracted from videos into a Residual Network. The second stream uses optical flows that capture motion between frames as input to a similar neural network. Each of the stream can independently output a binary variable indicating whether fall action occurs in each video. With the fusion of the two streams, the model should be able to capture both spatial and temportal information.

2 Related work

Earlier work on video-based human activity recognition has proposed methods based on shallow high-dimensional encodings of local spatial-temporal features. The algorithm in [9] describes sparse spatial-temporal interest points using local spatial-temporal features, which are then encoded into the Bag Of Features (BoF) representation. By pooling the BoF over several spatial-temporal grids, this method combines the representation with an SVM classifier. However, sparse interest points have been outperformed by dense sampling of local features [10]. shallow video representations [11] use dense point trajectories computed with optical flow to adjust local descriptor support regions. The trajectory-based pipeline achieved best performance by Motion Boundary Histogram [12], a gradient-based feature computed on both vertical and horizontal components of optical flow.

Later works have applied deep neural network architecture for video activity recognition. Using a stack of consecutive video frames as input, the model is supposed to automatically learn both spatial and temporal patterns only based on a sequence of images. Some architectures include HMAX model [13] and convolutional RBM and ISA in a discriminative model [14]. However, many of these models are proved not to well capture motion in videos, and some actually yield low accuracy than trajectory-based representation.

More recent works develop a two-stream ConvNet, where the temporal stream is based on dense optical flow between consecutive frames represented by vector field [15], which captures the motion of each pixel video frames [1]. By integrating the results of the spatial stream and temporal stream by averaging and SVM and training on UCF-101, the algorithm achieved higher accuracy than trajectory-based methods and spatio-temporal HMAX network.

3 Dataset and Features

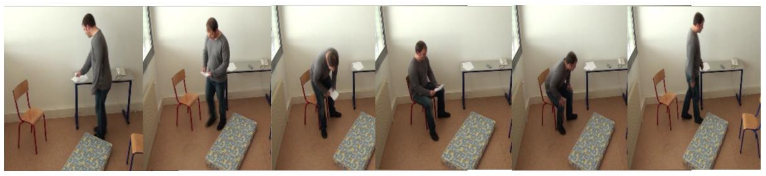


Figure 1: Sample Video of "No Fall"

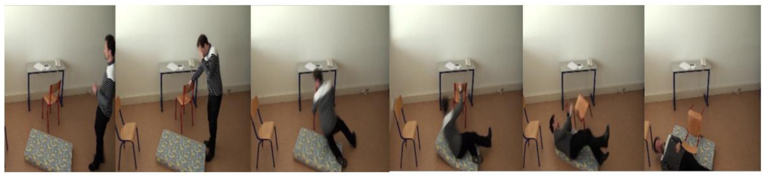


Figure 2: Sample Video of "Fall"

Our dataset of fall videos comes from three online sources: UR Fall Detection Dataset [16], Le2i [17] Dataset, and Multiple Camera Fall Dataset [18]. These videos are taken with surveillance cameras in a variety of background settings, including home, coffee room, office, and lecture room. Each setting contains both fall videos and "no-fall" videos. Among the total 873 videos, 427 of them contain fall actions.

Since fall is only one of the various kinds of human activities, the number of fall videos available online is limited. We mitigate this problem by cutting long videos with multiple falls and other actions into multiple shorter clips and applying data augmentation methods such as flipping, rotation, and rescaling. Furthermore, we use pre-trained models for ImageNet. Although the modes are not trained based on human activities, they do capture many high level details of pictures. The transfer learning allows us to train the recognization model with less data and fewer epoches. However, the number of fall videos is still a limiting factor for this project.

4 Methods

As mentioned previously, we apply a two-stream convolutional neural network. The first stream capture the spatial information with individual frames as input, which is used to detect scenes and objects in the videos. The seoncd stream is in the form of motion across frames, which carries information of the movement of the observer (the camera) and the objects.

4.1 Spatial stream ConvNet

The spatial stream, which uses individual video frames as input, extracts information from still images. It detects static features, since most actions are highly associated with particular objects. For our project, it should capture position and pose of humans in the video. This stream is based on

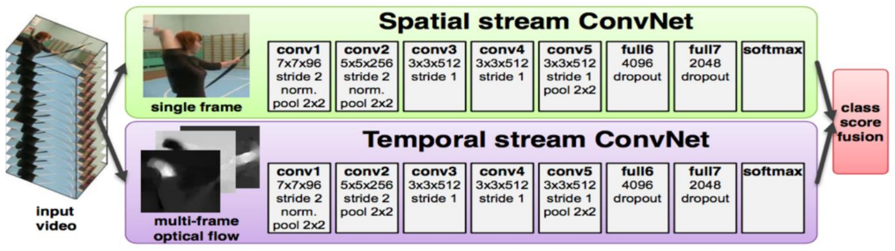


Figure 3: Architecture of Two-Stream Video Classification [4]

the assumption that people fall down if there is someone lying on the groud. We built our model upon recent advanced and popular recognition methods (ResNet18, ResNet101 and ResNet 152), and pre-trained the network on a large image classification dataset – ImageNet challenge dataset.

4.2 Temporal stream ConvNets

Unlike spatial stream, the input of temporal recognition stream is stacking optical flow displacement fields between several consecutive frames. The detection of motion between consecutive frames can extract more information from the videos and make the recognition easier and more accurate. Figure 4 (c) is a close-up dense optical flow in the outlined area. Figure 4 (c) and (d) are the horizontal d^x and vertical component d^y respectively, where higher intensity corresponds to positive values and lower intensity corresponds to negative values. The input to our ConvNet contains flows for both x and y direction.

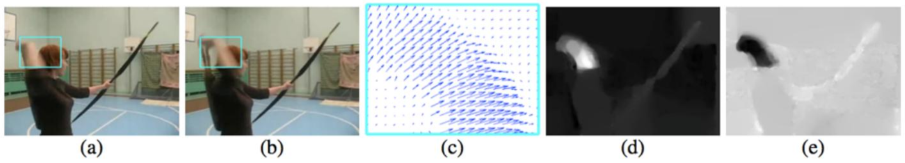


Figure 4: Optical Flow

4.3 Implementation Details

Frame Extraction and data pre-processing.

We implemented frame extraction by using FFmpeg with the ratio 10 frames per second. Since ’fall’ is an action that only last for several sections, a long video, like longer than 30s, will contains many useless information, which will influence our recognition and increase our computation. Therefore, we cut each video into 10 to 20 seconds length. This also increases the amount of data that people are not falling. To generate more data for fall activities, we also apply rotation and scaling. More details can be found in the code.

Computing Optical Flow

Optical flow captures the motion between two consecutive video frames. As shown in Figure 4, to compute optical flow, we first convert two frames (a) $I_0(x)$ and (b) $I_1(x)$, with $x = (i, j)$ the pixel index , to a vector field (c) in which each pixel has a two-dimensional vector indicating the motion of this pixel. Then the horizontal components of the vector field are transformed to a grayscale image (d), the vertical components to image (e). Image (d) and (e) are the optical flows we constructed. They are used as input for the temporal model. In implementation, we use OpenCV to realize TV-L1 optical flow estimation.

Training.

We mainly implemented [1] and [3] during the training procedure. The training procedures are similar for both spatial and temporal nets. we used mini-batch stochastic gradient to learn each parameter. The size of mini-batch is 25 while we use 250 epoches. For each video, spatial stream randomly samples frames from the second half part, because almost all the videos people fall at the second half part. In spatial net training, we randomly crop a 224×224 sub-image from select frame;it then undergoes random horizontal flipping and RGB jittering. Then videos are rescaled so that the smallest side of the frame equals 256. In the temporal net, we selected 10 frames based on the length of the video. We try to cover the motion of the whole video with a focus on second half. Detailed implementation can be found in the code. Then we computed optical flow volumes I for the selected frames. Currently, our input is a fixed-size $224 \times 224 \times 20$, which undergoes randomly cropped and flipped. We initialized our learning rate to be 10^{-2} which decays to 1/10 after 100 iterations . We set regularization dropout ratio to 0.8 to avoid over-fitting and improve our performance.

Testing.

For the test set, we apply a slight different sampling method compared to training. We sampled 25 times for the same video, and run the model for these samples and finally average the prediction results. For example, for the temporal stream we randomly select 10 consecutive frames from the video and run the model, and repeat this process for 25 times and predict the label based on the average result.

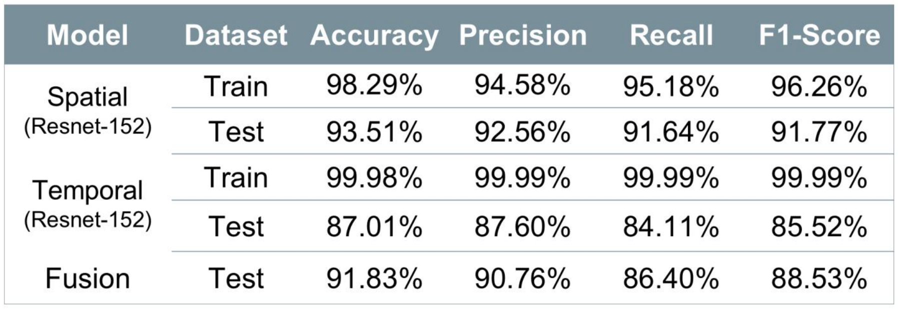
Fusion

Due to time limit, we only implemented one fustion method on test set. We extract the last layer information (the input for softmax) for spatial and temporal stream on the test set. We take the mean value and then apply softmax. In the future, we should incorporate the fusion in the training process to take lagest advantage of both temporal and spatial information.

Learning framework and Multi-GPU training.

We implemented our results based on PyTorch framework. We included parallel training on multiple GPUs. We are sponsored by China Mobile with 3 NVIDIA Titan X cards, which great shorten our training time.

5 Results and Discussion

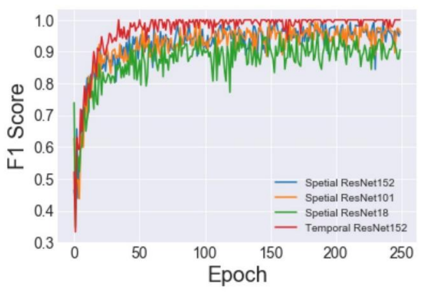


Model	Dataset	Accuracy	Precision	Recall	F1-Score
Spatial (Resnet-152)	Train	98.29%	94.58%	95.18%	96.26%
	Test	93.51%	92.56%	91.64%	91.77%
Temporal (Resnet-152)	Train	99.98%	99.99%	99.99%	99.99%
	Test	87.01%	87.60%	84.11%	85.52%
Fusion	Test	91.83%	90.76%	86.40%	88.53%

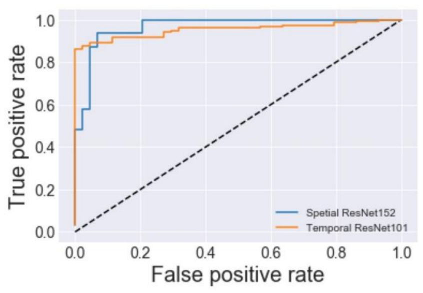
Figure 5: Performance of Different Networks

Figure 5 and Figure 6 shows parts of our results. Figure5 compares the performance using ResNet-152 in the Spatial and Temporal Stream. Both spatial and temporal stream have good accuracy and F1-score. Spatial stream's F1-score is 96.26%, and temporal stream is even as high as 99.99%. On the test set, the spatial stream still performs good with 91.77 F1-score and 93.51% accuracy. As for the temporal stream, the test set go down to 85.52% of F1-score and 87.01% of accuracy. It is obviously to be over-fitting on the Temporal Stream. This is the result that after adjustment using early stopping strategy. So our next step is trying figure out how to reduce the prediction bias on train and test set. Since our final result will both consider spatial feature and temporal feature, this might be a good method to prevent overfitting and have a good prediction. After fusion, the final accuracy is 91.83%, which is lower than that of spatial only. This indicates that the two stream model performance is hold back by the temporal stream. Figure 6a implies the learning paths of Spatial ResNet-18,101,152 and Optimal Flow ResNet-152.

By examining the videos on which our model makes incorrect predictions, we discover that our model fails to detection falls (1) when the person falls on an object that is not the ground (e.g. falling



(a) Training Set F1 Score



(b) Test Set Roc Curve

Figure 6: Training and Test Set Performance

on a coffee table) and (2) when the action of falling happens within a very short time in a relatively long video. The first problem can be addressed by gathering more videos with people falling on objects rather than on the ground, or adding weights on these videos during training process. The second issue results from the limited number of frames capturing fall action. This can be improved by adjusting the speed of some videos or the frame rate.

Another problem is that false negative rate is higher than false positive rate, which means our model is more likely to predict a fall video as "no fall" than the other way around. Since the failing to detect fall occurrence has much more severe consequences than sending a wrong alarm when there is no falling, our model should put higher priority on minimizing false negative rate. In training and prediction, we can set the classification probability threshold lower than the default 0.5 to control the false negative rate.

6 Conclusion and Future Work

By applying two-stream ConvNet model, we proposed a video-based fall detection model with a competitive result (F1 score of 91.77%). Currently, the spatial stream performs much better than temporal stream on test set. Improving temporal stream is essential to yield better result. Also, better fusion method is needed for both training and testing process. Futhemore, there are a few other things we can do in the future:

1. Solve the overfitting problem for Temporal streams. Some strategies we are going to implement include finding and generating more data, and testing different models. At least $10K$ videos would be very helpful for our model.
2. Implement human detection before activity recognition. This will exclude most of background noises, thus allow our model to better perform on videos taken in diverse background settings and situations. This improvement also makes it possible to detect falls in scenes with multiple people.
3. Fall detection is critical to human safety and has a wide range of applications in reality and huge market potential. How to transfer our model in product might be a final goal of this project. Besides improving the prediction performance, how to make the model small and fast should be another key points that need to be solved.

7 Contributions

Each team member contributes equally to the project. Jie and Weixuan are in charge of the searching for resources, coding and execution. Qianmin takes care of data collection, video clipping, labeling, and error analysis. We have worked together on analyzing the results, making the poster and final write-up, and regular communication with our sponsor China Mobile. We are happy with each other's contribution.

References

[1] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." In Advances in neural information processing systems, pp. 568-576. 2014.

[2] Solbach, M. D., and Tsotsos, J. K. Vision-Based Fallen Person Detection for the Elderly. arXiv preprint arXiv:1707.07608. 2017.

[3] Yi Zhu, PyTorch implementation of popular two-stream frameworks for video action recognition, retrieved from https://github.com/bryanyzhu/two-stream-pytorch, March 2018.

[4] J. A. Stevens, P. S. Corso, E. A. Finkelstein, & T. R. Miller. The costs of fatal and non-fatal falls among older adults. Injury Prevention, 12(5):290–295, 2006.

[5] Bogdan Kwolek, Michal Kepski, Human fall detection on embedded platform using depth maps and wireless accelerometer, Computer Methods and Programs in Biomedicine, Volume 117, Issue 3, December 2014, Pages 489-501, ISSN 0169-2607. http://fenix.univ.rzeszow.pl/ mkepski/ds/uf.html

[6] Ma, C.-Y., Chen, M.-H., Kira, Z., & AlRegib, G. (2017). TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition. Retrieved from http://arxiv.org/abs/1703.10667

[7] E. Auvinet, C. Rougier, J.Meunier, A. St-Arnaud, J. Rousseau, "Multiple cameras fall dataset", Technical report 1350, DIRO - Université de Montréal, July 2010.

[8] I. Charfi, J. MitÈran, J. Dubois, M. Atri, R. Tourki, "Optimised spatio-temporal descriptors for real-time fall detection: comparison of SVM and Adaboost based classificationî, Journal of Electronic Imaging (JEI), Vol.22. Issue.4, pp.17, October 2013.

[9] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In Proc. CVPR, 2008.

[10] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In Proc. BMVC., pages 1–11, 2009.

[11] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In Proc. CVPR, pages 3169–3176, 2011.

[12] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In Proc. ECCV, pages 428–441, 2006.

[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In Proc. ICCV, pages 2556–2563, 2011.

[14] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In Proc. CVPR, pages 3361–3368, 2011.

[15] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In Proc. ECCV, pages 25–36, 2004.

[16] Michal Kępski. UR Fall Detection Dataset. URL: http://fenix.univ.rzeszow.pl/ mkepski/ds/uf.html

[17] Antoine Trapet. Fall Detection Dataset. URL: http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr

[18] Multiple Camera Fall Dataset. URL: http://www.iro.umontreal.ca/ labimage/Dataset/