

TOWARDS MUSCULOSKELETAL SIMULATION-AWARE FALL INJURY MITIGATION: TRANSFER LEARNING WITH DEEP CNN FOR FALL DETECTION

Haben Yhdego
Jiang Li
Steven Morrison
Michel Audette

Old Dominion University
5115 Hampton Boulevard
Norfolk, VA, USA

{hyhde001, jli, smorriso, maudette}@odu.edu

Christopher Paolini
Mahasweta Sarkar

San Diego State University
5500 Campanile Drive
San Diego, CA, USA
{cpaolini, sarkar2}@sdsu.edu

Hamid Okhravi

Eastern Virginia Medical School
825 Fairfax Ave
Norfolk VA, USA
okhravhr@evms.edu

ABSTRACT

This paper presents early work on a fall detection method using transfer learning method, in conjunction with a long-term effort to combine efficient machine learning and prior personalized musculoskeletal modeling to deploy fall injury mitigation in geriatric subjects. Inspired by the tremendous progress in image-based object recognition with deep convolutional neural networks (DCNNs), we opt for a pre-trained kinematics-based machine learning approach through existing large-scale annotated accelerometry datasets. The accelerometry datasets are converted to images using time-frequency analysis, based on scalograms, by computing the continuous wavelet transform filter bank. Subsequently, data augmentation is performed on these scalogram images to increase accuracy, thereby complementing limited labeled fall sensor data, enabling transfer learning from the existing pre-trained model. The experimental results on publicly available URFD datasets demonstrate that transfer learning leads to a better performance than the existing methods in the case of scarce labeled training data.

Keywords: Fall Detection, Deep CNN, Transfer Learning, Data Augmentation, Continuous Wavelet Transform (CWT).

1 INTRODUCTION

There are over 46 million senior citizens of age 65 in the US, including 1.4 million living in nursing facilities (NFs) and assisted living facilities (ALFs), for whom falls are the leading cause of injury-related death and hospitalization. Every year, around 30% of seniors living in the community and 50% of seniors

living in residential care facilities experience one or more falls, and half of them experience multiple falls (MQ, Weintraub, and Rubenstein 2004). The elderly are exposed to falls as a result of the deterioration of their physical functions (Park, Shin, and Ha 2009). The reasons why falls occur and why some people are more likely to fall may be intrinsic or extrinsic. Intrinsic factors equate with chronic conditions such as Alzheimer, Parkinson, arthritis, dementia and visual impairment. Conversely, extrinsic factors are environmental hazards, which include slippery surfaces and dangerous furniture configurations. Fall events may result in severe physical injuries as well as psychological issues, depression and avoidance of the activity that caused the fall.

This dire situation has led to the development of automatic fall-detection systems (Lim et al. 2014), the long-term goal of which is to trigger deployment of injury-mitigation mechanisms, such as an airbag. In conjunction with these developments, fast and early fall detection is essential to the timely deployment of pneumatic protection actuation. The identified fall-detection systems can be classified primarily into two categories: context-aware systems and wearable devices. Most of the wearable automatic sensors utilize accelerometers, gyroscopes, or tilt sensors and set thresholds of their outputs to detect the large impact of body with ground or near horizontal orientation of trunk, or both (Lim et al. 2014).

The high incidence and dire implications of fall events in the geriatric population motivate our nascent project on fall detection, the long-term goal of which is to combine machine learning (ML) that keys on real-time kinematics with a complementary pre-conditioner that encodes clinical subject-specific prior factors, through fuzzy logic for example. The role of simulation in this architecture will center on the application of personalized dynamic musculoskeletal simulation to anticipate the normal gait of a subject to better recognize deviations from this norm, which are likely to equate with a fall event, based on open-source tools such as [OpenSim](#) (Hicks and Delp 2010). The personalization of OpenSim is feasible through the combined application of commercial range-sensing (such as the iPad-mounted Structure sensor [Structure](#) (Occipital 2016)), which will produce a 3D point-set that encodes the outer shape of each geriatric subject, as well as a detailed multi-surface model of the anatomy, found in [BodyParts3D](#) (N. Mitsuhashi 2009)), and deformable multi-surface models (Haq et al. 2015, Gilles and Magnenat-Thalmann 2010), which typically warp an anatomical model to MRI or CT but can also map it to 3D range data. In addition to dynamic simulation-aware considerations, the preconditioner will also ratchet up or down the sensitivity of the detection, much like a downhill skier sets the tension on his bindings according to his/her skill level, to account for static intrinsic factors that make a fall more likely, such as dementia. This paper describes the machine-learning component of the proposed software architecture, which will ultimately be potentiated by the simulation-aware pre-conditioning component.

Deep convolutional neural networks (DCNN), characterized by multiple intermediate “hidden” layers, have gained popularity due to their ability to recognize complex patterns in image and video for classification, while providing highly nonlinear parameter-space boundaries that enable sophisticated decisions. However, there has been a paucity of formal understanding on how to interpret those patterns, why they perform well and how they might be optimized. Researchers typically used trial-and-error, by tuning variables for boosting their performance. As DCNN is not a black box, we can visualize features learned to understand the intermediate representation of the transformed data in the network. After applying the forward DCNN for the images, these intermediate forms are themselves images, and can be visualized as images. Recently, the “Deconvolution approach” (Zeiler and Fergus 2013) was proposed to visualize features on the hidden layers by projecting those feature activation’s back to the input pixel space (images). We can transfer the knowledge learned from the early network of existing large datasets to other datasets by taking the lower-level features that are generic to any datasets (like blobs and edges). This work describes a fall detection algorithm using transfer learning and continuous wavelet analysis of tri-axial accelerometer fall sensor data to classify fall and non-fall datasets by using the pre-trained CNN architecture Alexnet. Wavelet-based time-frequency representations of sensor data are used to create scalograms as RGB images to be compatible with our Alexnet architecture, which also facilitates visualization, where each of its RGB image input should be

an array size of 227 by 227 by 3. Data augmentation on these images is performed to increase the size of the dataset used to fine-tune the Alexnet deep CNNs.

2 RELATED WORKS

Activity and pattern recognition has elicited academic research and industry over the last decades. Fall detection is an example of activity recognition that has garnered interest; several methods have been developed with varying levels of accuracy (Lim et al. 2014, Edy, Putray, and Vesilo 2017, Bourke and Lyons 2008, Kwolek and Kepski 2014, Purwar, Jeong, and Chung 2007). Some use only wearable inertial sensors, while others combine cameras and ambient sensors deployed in a living environment. Methods of fall detection vary, however the most common implementations involve either thresholding or machine learning (Figueiredo et al. 2016).

In the simple threshold method, threshold values of specific parameters calculated from sensor data such as tri-axial acceleration are used to detect a fall. Automatic fall detection using a threshold-based method of single parameters, calculated from accelerometry measurements, has a high sensitivity; however, it has a relatively low specificity (Kangas et al. 2009). For instance, Purwar used a tri-axial accelerometer to set thresholds of acceleration and orientation of the torso through experiments to detect falls, which achieved an accuracy of 81% (Purwar, Jeong, and Chung 2007). Alternately, in order to calculate the torso's angular position and acceleration prior to thresholding, Bourke implements a bi-axial gyroscope to detect fall events (Bourke and Lyons 2008).

Meanwhile, Shi et al. proposed an airbag system to protect hip during falls, in a manner comparable to airbags deployed in a car crash (Shi et al. 2006, Shi et al. 2009). To ensure that the airbag was fully filled before a collision they applied a tri-axle accelerometer and gyroscope, then set accelerations and angular velocities for thresholding and use Support Vector Machine (SVM) methods as classifier to detect the fall events. However, accuracy and throughput were not assessed (Tong et al. 2013).

Other approaches, like (Dai et al. 2010) and (Lee and Carlisle 2011), proposed thresholding both the magnitude and the vertical projection of the acceleration. As a result, when both thresholds are exceeded within a time window, a fall is detected. The latter paper by (Lee and Carlisle 2011) uses angular velocities in addition to acceleration magnitude as input data. When both values are greater than the specified thresholds a fall is detected. In a related work, Viet uses both upper and lower thresholds for the acceleration magnitude (Viet and Choi 2011). If the upper and lower is exceeded in less than 1s, a fall template using wavelet transform is used for comparison with the acceleration signal. A fall is detected if the comparison produces a high similarity value. Even though most of these methods depend on thresholds, it is difficult to use threshold methods to detect unseen data. Due to this limitation, recent approaches (Ozdemir and Barshan 2014) employ feature extraction engineering and machine learning classifiers to improve detection accuracy.

The machine learning implementations differ in many aspects. Some of them like (Abbate et al. 2012) and (Viet and Choi 2011) combine both thresholding and machine learning methods. Tong's work (Tong et al. 2013) uses wearable sensors to collect the data, non-overlapping sliding windows to extract the features, and combined Hidden Markov Models (HMM) and thresholding to classify it. This result shows perfect specificity (true positive rate) and sensitivity (true negative rate). Even though, HMM is a sequential model that takes advantage of time series data of fall events, the result does not indicate that the algorithm is sufficiently efficient to be used in real time. Also, because the data collected is not real fall situations of geriatric subjects, but simulated data from youthful subjects. Moreover, the dataset does not contain near-fall events. Many fall detection algorithms suffer in classifying near-fall and fall events. Furthermore, Tong's method employs thresholding to boost the performance; however, thresholding cannot cope with unseen data and it will wrongly classify near-fall events in realtime fall detection.

Nowadays, several machine learning algorithms are used to further improve the detection rate accuracy. There are many ways to increase the accuracy of fall detection; the best-known improvements are to collect the data and extract features carefully and to select optimal algorithm for classification. Extracting features from the sensor signals that are inputs to the machine learning techniques has an impact on the accuracy of the detection algorithms. There exist many types of feature extraction methods of signal data; the sliding window is the most widely used method. Several studies show how window size and type impact the detection rate. A method proposed by Edy (Edy, Putray, and Vesilo 2017) compares the impact of fixed-size non-overlapping sliding window (FNSW) and fixed-size overlapping sliding window (FOSW) for fall detection. This study uses two publicly accessible datasets: the Cogent and SisFall datasets (Ojetola, Gaura, and Brusey 2015, Sucerquia et al. 2017) collected with wearable sensors, while supervised machine learning algorithms of K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) are used as a classifier. The results differ according to different datasets: The FOSW does better with the Sisfall dataset (Sucerquia, Lopez, and Vargas-Bonilla 2017) and FNSW performs better with the Cogent datasets (Ojetola, Gaura, and Brusey 2015). In addition, Putu does not assess the effect of window size and type on the performance of each classifier.

Recently, deep neural networks (like CNN and RNN) have been employed for time series classification of human activity recognition tasks. To train deep neural networks like alexnet from scratch, we need large dataset like the UCL human activities dataset (Jorge et al. 2015). But, it is difficult to get enough elderly fall data-set for training deep neural networks from scratch. Hence, it is known that a model pre-trained from generic features extracted and learned from existing large datasets may confer advantages in applications with less data. To deal with this limitation, we propose to apply transfer learning from deep CNN methods for classifying our wearable sensors data, which helps to cope with unseen data. We also apply the continuous wavelet transformation to convert the sensor data to RGB images, suitable for input of our Alexnet architecture.

3 METHODS

A publicly available UR fall detection dataset [URFD](#) (Kepski 2014, Kwolek and Kepski 2014) has been used. This dataset contains 70 sequences of which 30 of them involve falls and 40 of them consist of non-fall activities. Kepski et al recorded these datasets from two accelerometers placed at the waist and depth images collected using two Microsoft Kinect cameras. For our proposed method, we use the accelerometer data only, which consists of 3D acceleration data $\mathbf{a}(t) = [a_x(t), a_y(t), a_z(t)]$ and also the magnitude of these 3D acceleration vectors is computed as follows:

$$S_v(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)}.$$

The data is sampled at a rate of 256 HZ. These data points are preprocessed using median filtering with window size of 10 to remove the high-frequency noise. We can see in figure Figure 1 and Figure 2 the sample waveforms after performing the preprocessing for the fall and non-fall datasets respectively.

Due to the beneficial impact of additional training data on model generalization and performance (Taewoong et al. 2017), the effectiveness of a data augmentation procedure involving random rotations has been used as well. Moreover, data augmentation introduce label-invariant variability of wearable sensor data (Taewoong et al. 2017). For example, an upside-down placement of the sensor can invert the sign of the sensor readings without changing the labels. Therefore, augmentation by applying arbitrary rotations to the existing data can be used as a way of simulating different sensor placements. Given the acceleration vector $\mathbf{a}(t) = [a_x(t), a_y(t), a_z(t)]$ at time t , which contains the acceleration components along the x , y and z axes of

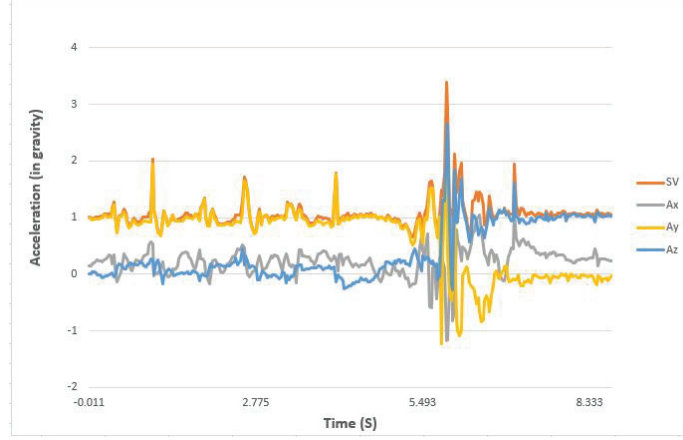


Figure 1: Fall dataset.

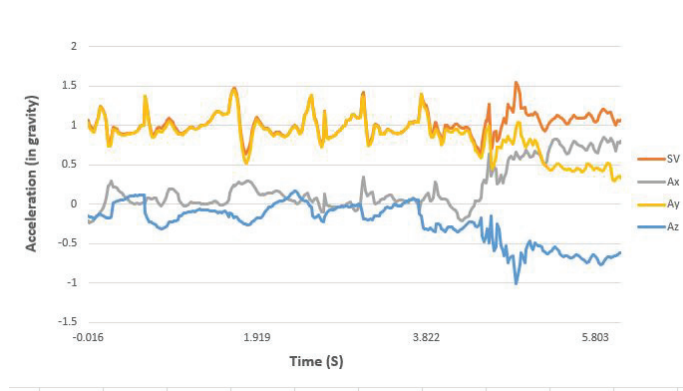


Figure 2: Nonfall dataset.

the device respectively, a new vector $\mathbf{a}'(t)$ can be obtained by rotating $\mathbf{a}(t)$ by 10 degree in the x,y and z axis is performed.

3.1 Continuous Wavelet Transform

Ravi et al (Ravi et al. 2016) show the importance of using a suitable domain when a deep learning methodology is applied to time-series data. Specifically, they show that the scalogram representation is essential for extracting interpretable features that capture the intensity differences among nearest inertial sensor data points. The scalogram representation also provides a form of time and sampling-rate invariance. This enables the classification to be more robust against data shifting in time and against changes in signal amplitude and sampling rate. Moreover, frequency selection in the scalogram domain also provides an implicit way to allow noise filtering of the data over time.

To use transfer learning to our fall datasets, we must transform the 1D signal data to RGB images(three channel input). This conversion is necessary because our transfer learning approach exploits the feature vectors from the pretrained model of the Imagenet datasets (Deng et al. 2009) and Alexnet architecture is designed to classify images (three channel inputs). We reuse the Alexnet network architecture of the DCNN to classify accelerometer signals based on scalogram images from the continuous wavelet transform (CWT) of the time series data. Mathematically, CWT represents time-varying spectral information of a continuous

signal by computing the inner products of this signal with a set of continuous wavelets. To create time-frequency representations of the accelerometer signals of the wearable sensor data we have to pre-compute a CWT filter bank. Pre-computing the CWT filter bank is the most computationally efficient method when obtaining the CWT of many signals using the same parameters. These representations are aggregated into a scalogram, which summarizes the magnitude of the CWT coefficients of a signal. Our inputs to the Alexnet architecture are the scalograms RGB images as shown in figure Figure 3. To be compatible with Alexnet, each RGB image is transformed to an array of size 224-by-224-by-3.

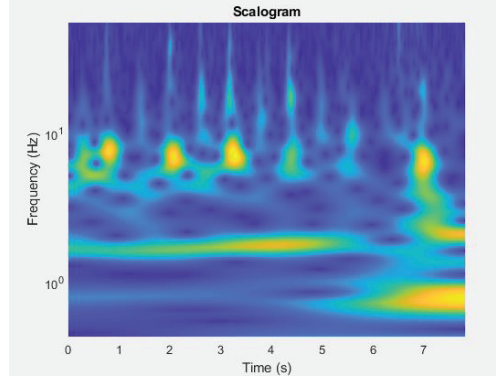


Figure 3: Sample Scalogram.

3.2 Transfer Learning with Deep CNN

In practice, we don't usually train an entire DCNN from scratch with a random initialization. Indeed, it is relatively rare to have a sufficiently large dataset for the depth of network required. Instead, it is recommended to pre-train a DCNN on a very large dataset and then use the trained DCNN weights either as an initialization or a fixed feature extractor for the task of interest, which we call transfer learning. The low and high-level features learned by a CNN on a source domain can often be transferred to augment the learning in a different target domain. For target problems with a large dataset, we can transfer the low-level features (such as edges and corners), and learn new high-level features specific to the target problem. For target problems with limited amounts of data, learning new high-level features is difficult. However, if the source and target domains are sufficiently similar, the feature representation learned by the CNN on the source task can be transferred to the target problem. Deep features extracted from CNNs trained on large annotated datasets of images have been used as generic features very effectively for a wide range of computer vision tasks (Donahue et al. 2013, Oquab et al. 2014).

AlexNet is a deep CNN whose architecture supports images of size 227 by 227 with three channels (RGB) as shown in figure Figure 4. The RGB images of accelerometry data of the ax, ay, az and Sv are the inputs to our architecture. Fine-tuning the transferred model allows us to bring a significant improvement of average accuracy for our fall detection classification with insufficient data. We fine-tune the weights of the pre-trained DCNN of imagenet by continuing the back-propagation. It is possible to fine-tune all the layers of the DCNN, or it's possible to keep some of the earlier layers fixed (due to overfitting concerns) and only fine-tune some higher-level portion of the network. This is motivated by the observation that the earlier features of a DCNN contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks, but later layers of the DCNN becomes progressively more specific to the details of the classes contained in the fall images dataset. As our fall detection dataset is small, we can train a linear classifier for the feature vectors learned from the Imagenet dataset. However, the RGB images of our fall datasets are very different from the Imagenet dataset, so it is better to train the architecture with our fall

datasets earlier in the network than applying linear classifier at the end layer of the network, which contains more input-specific features of Imagenet. Consequently, the last three layers of the AlexNet architecture are changed and retrained for classifying the RGB image of our fall dataset. Those three layers must be fine-tuned to our fall classification problem: the fully connected layer, which is configured for 1000 categories by default, must be set to the same size as the number of categories in our fall dataset, which is two classes in our case. The Softmax classifier, which provides a probability for each class in a multi-class problem, is also replaced with the support vector machine(SVM) classifier that classifies fall and non-fall events in a straightforward manner (Trivedi 2016). Furthermore, the SVM classifier provide stable results and trains faster than the Softmax classifier.

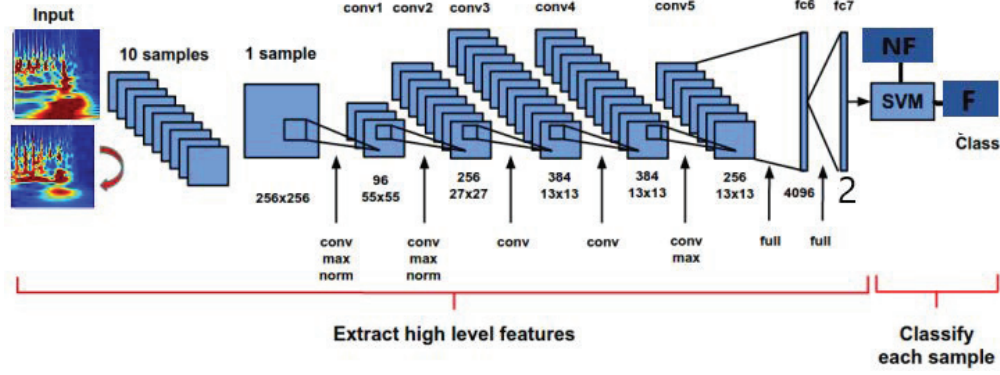


Figure 4: Proposed Alexnet Architecture.

4 EXPERIMENT RESULTS

The URFD datasets is divided into training and testing with 80% and 20% respectively. After we train and test on the higher layers of the proposed alexnet architecture above, our training progress over the training iteration number is shown in figure Figure 5. The experimental evaluation on the URFD dataset gives us the confusion matrix shown in Figure 6 and results show in the table Table 1 based on the formulas $Accuracy = \frac{TP+TN}{TP+TN+FN+FP}$, $Precision = \frac{TP}{TP+FP}$, $Sensitivity = \frac{TP}{TP+FN}$, and $Specificity = \frac{TN}{TN+FP}$, where TP is True positive, TN is True Negative, FN is false negative and FP False positive.

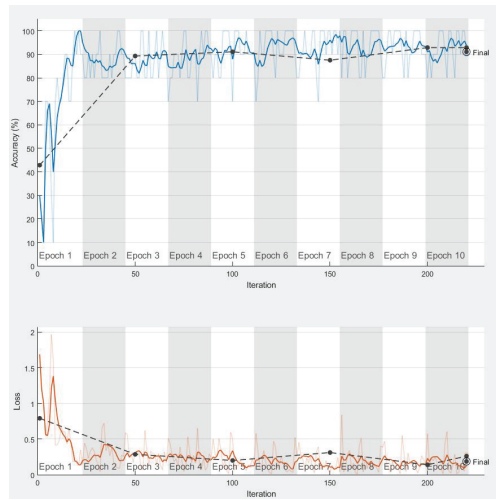


Figure 5: Accuracy and loss over the training iteration.

Table 1: Comparing the results of different methods.

Metrics	SVM	TB	TL(Our Method)
Accuracy	92.86	92.86	96.43
Precision	94.17	90.00	95.83
Sensitivity	90.0	96.67	95.83
Specificity	95.0	90.0	96.875

The Threshold Based and Support Vector Machine methods are compared with our Transfer Learning(TL) method. The first method was proposed in (Bourke and Lyons 2008) and is denoted as (TB). It uses a threshold on the acceleration magnitude in order to detect the fall. In training, the threshold is determined as the minimum of the magnitude peaks during the fall happens. In testing, the same threshold is used for detecting fall from non-fall. Where-as, the second method, SVM was proposed in (Kwolek and Kepski 2014) and originally used both accelerometer data and Kinect depth data. It initiates the fall detection procedure by using a threshold value of 3g on the acceleration magnitude. This fall detection procedure consists of an SVM model trained on the extracted Kinect depth features that helps to make the final decision (Theodoridis et al. 2017). The depth features were scaled so that they have zero mean and unit variance.

Output Class	F	23 41.1%	1 1.8%	95.8% 4.2%
	NF	1 1.8%	31 55.4%	96.9% 3.1%
		95.8% 4.2%	96.9% 3.1%	96.4% 3.6%
		Target Class		

Figure 6: Confusion Matrix.

The TB method has the highest amount of false positives (detecting a fall when no fall has occurred), as it has the lowest specificity (90%). It has the same accuracy with the SVM method, which however has higher specificity, but lower sensitivity. This result means that the SVM method produces fewer false positives, but at the dire cost of more false positives. Meanwhile, the proposed Transfer Learning (TL) method is better than the previous methods in all evaluation metrics except the sensitivity with the TB method (which is 96.67% for TB and 95.0% for TL method). Even though the transfer learning relies only on the accelerometer data, it has only produced a single false positive result (specificity 96.875%), while the sensitivity of 95.83%. Moreover, the TB and SVM methods are invariant to rotations as they didn't use data augmentation used in our transfer learning method.

With Deep CNN, visualization has been used in computer vision as a tool in feature evaluation. Visualization of image features helps to understand the failurity of models (Pirsiavash et al. 2013). Each layer of a CNN produces a response, or activation, to an input image. These layers of a CNN consist of many 2D arrays. We can examine the corresponding activations and discover which features AlexNet learns best by comparing areas of activation with the original image and by passing the image through the network to examine the output activations of the convolutional layer. Visualizing the convolution layers with top neurons that learned

from our fall datasets helps us to visualize what we learned from the model and to diagnose its potential problems of the model by looking at to the visualized features. The strong activation projected separately down to pixel space as shown in figure Figure 7 shows that our fall datasets were learned well by our model.

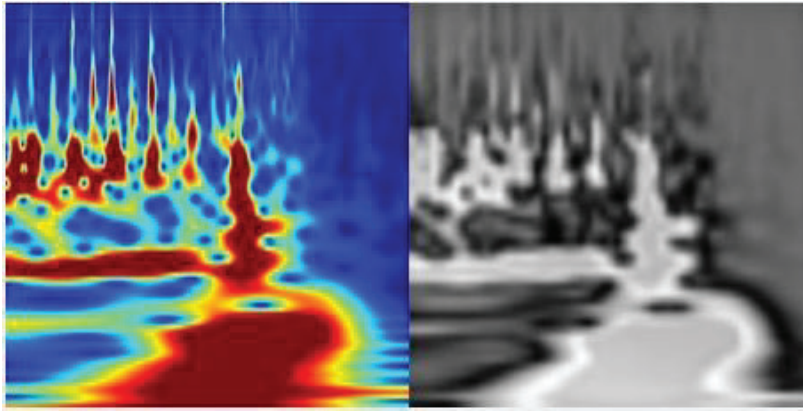


Figure 7: Sample input image and its visualization of the learned feature of top-activated neuron in roughly aligned layers (right before Fully Connected (FC) layers).

5 CONCLUSION

This paper describes a fall detection methodology that combines transfer learning and continuous wavelet analysis of tri-axial accelerometer sensor-based data to classify fall and non-fall data-sets, in conjunction with pre-trained CNNs based on the Alexnet architecture. The TL method is used for the purpose of overcoming the difficulties of training a deep CNN resulting from limited fall datasets by transferring knowledge learned from a large amount of existing Imagenet datasets (Deng et al. 2009). Wavelet-based time-frequency analysis represents the sensor data by scalograms, whereby RGB images of those scalograms are then suitable for input into the Alexnet deep learning framework. Our method uses the accelerometer data only but the results are better than the competing methods that uses both the Kinect and accelerometer data. In addition, data augmentation is beneficial for improving the classification performance to some degree.

Detecting timely falls of the elderly is significant to their safety. In our future work, we are going to work on a fall detection system that uses the acceleration sensor to detect a senior citizen falling in real time fall detection scheme. In addition, we anticipate that integration with a pre-conditioner that encodes personalized simulation-based dynamic gait priors and intrinsic bias factors will further improve performance.

REFERENCES

- Abbate, S., M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio. 2012. "A smartphone-based fall detection system". *Pervasive and Mobile Computing*.
- Bourke, A. K., and G. M. Lyons. 2008. "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor". *Med. Eng. Phys.* vol. 30, pp. 84–90.
- Dai, J., X. Bai, Z. Yang, Z. Shen, and D. Xuan. 2010. "PerFallID: A pervasive fall detection system using mobile phones". *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications (PERCOM 2010)*.
- Deng, J., W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database.". *IEEE Computer Vision and Pattern Recognition (CVPR)*.

- Donahue, J., Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. 2013. "A deep convolutional activation feature for generic visual recognition." *DeCAF: arXiv:1310.1531*.
- Edy, P., S. Putray, and R. Vesilo. 2017. "Window-size impact on detection rate of wearable-sensor-based fall detection using supervised machine learning". *IEEE*.
- Figueiredo, I. N., C. Leal, L. Pinto, J. Bolito, and A. Lemos. 2016. "Exploring smartphone sensors for fall detection". *The Journal of Mobile User Experience* vol. 5.
- Gilles, B., and N. Magnenat-Thalmann. 2010. "Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations." *Med Image Anal.* vol. 14(3), pp. 291–302.
- Haq, R., J. Cates, D. A. Besachio, R. C. Borgie, and M. A. Audette. 2015. "Statistical Shape Model Construction of Lumbar Vertebrae and Intervertebral Discs in Segmentation for Discectomy Surgery Simulation". *CSI@MICCAI*, pp. 85–96.
- Jennifer Hicks and Scott Delp 2010. "OpenSim". <https://opensim.stanford.edu/>.
- Jorge, L., O. Reyes, A. S. Luca Oneto, X. Parra, and D. Anguita. 2015. "Transition-Aware Human Activity Recognition Using Smartphones". *Neurocomputing*. Springer.
- Kangas, M., I. Vikman, J. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and Jämsä. 2009. "Sensitivity and specificity of fall detection in people aged 40 years and over". *Gait e Posture* vol. 29, pp. 571–574.
- M. Kepski 2014. "UR fall detection dataset". <http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html>.
- Kwolek, B., and M. Kepski. 2014. "Human fall detection on embedded platform using depth maps and wireless accelerometer". *Computer Methods and Programs in Biomedicine* vol. 117, pp. 489–501.
- Lee, R. Y. W., and A. J. Carlisle. 2011. "Detection of falls using accelerometers and mobile phone technology". *Age and Ageing* vol. 40.
- Lim, D., C. Park, N. Kim, S. Kim, and Y. Yu. 2014. "Fall-Detection Algorithm Using 3-Axis Acceleration: Combination with Simple Threshold and Hidden Markov Model". *Journal of Applied Mathematics*.
- N. Mitsuhashi 2009. "BodyParts3D: 3D structure database for anatomical concepts". <http://lifesciencedb.jp/bp3d/#>.
- MQ, V., N. Weintraub, and L. Rubenstein. 2004. "Falls in the nursing home: are they preventable?". *J Am Med Dir Assoc.* vol. 5(6), pp. 401–6.
- Occipital 2016. "Structure Sensor: Structure by Occipital". <https://structure.io/>.
- Ojetola, O., E. Gaura, and J. Brusey. 2015. "Data set for fall events and daily activities from inertial sensors". *Proceedings of the 6th ACM Multimedia Systems Conference*, pp. 243–248.
- Oquab, M., L. Bottou, I. Laptev, and J.Sivic. 2014. "Learning and transferring mid-level image representations using convolutional neural networks." *CVPR*.
- Ozdemir, A., and B. Barshan. 2014. "Detecting Falls with Wearable Sensors Using Machine Learning Techniques". *Sensors* vol. 14, pp. 10691–10708.
- Park, B., I. Shin, and J. Ha. 2009. "Senior Survey 2008: life and welfare service needs of the elderly in Korea". *Ministry for Health and Welfare*.
- Pirsiavash, H., T. Malisiewicz, A. Torralba, C. Von-dricks, and A. Khosla. 2013. "Visualizing Object Detection Features." *Massachusetts Institute of Technology, Cambridge, MA 02139 USA*.
- Purwar, A., D. Jeong, and W. Chung. 2007. "Activity monitoring from real-time triaxial accelerometer data using sensor network". *IEEE Int. Conf. Control, Autom. Syst.*, pp. 2402–2407.
- Ravi, D., C. Wong, B. Lo, and G. Yang. 2016. "Deep learning for human activity recognition: A resource efficient implementation on low-power devices". in *Proc. 2016 IEEE 13th Int. Conf. Wearable Implantable Body Sensor Netw.*, pp. 71–76.

- Shi, G., C. S. Chan, W. J. Li, K.-S. Leung, Y. Zou, and Y. Jin. 2009. "Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier". *IEEE Sensors* vol. 9, pp. 495–503.
- Shi, G., C. S. Chan, Y. Luo, G. Zhang, W. J. Li, P. H. W. Leong, , and K. S. Leung. 2006. "Development of a human airbag system for falling protection using MEMS motion sensing technology". *IEEE Int. Conf. Intell. Robots Syst.*, pp. 4405–4410.
- Sucerquia, A., J. Lopez, and J. Vargas-Bonilla. 2017. "SisFall: A Fall and Movement Dataset". *Sensors* vol. 17, pp. 1.
- Taewoong, T., F. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulic. 2017. "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks". *ICMI*, DOI:10.1145/3136755.3136817.
- Theodoridis, T., V. Solachidis, N. Vretos, and P. Daras. 2017. "Human fall detection from acceleration measurements using a Recurrent Neural Network". *Information Technologies Institute, Centre for Research and Technology Hellas*.
- Tong, L., Q. Song, Y. Ge, and M. Liu. 2013. "HMM-Based Human Fall Detection and Prediction Method Using Tri-Axial Accelerometer". *IEEE SENSORS* vol. 13, pp. 5.
- Anusua Trivedi 2016. "Transfer Learning and Fine-tuning Deep Convolutional Neural Networks". <http://blog.revolutionanalytics.com/2016/08/deep-learning-part-2.html>.
- Viet, V., and D. Choi. 2011. "Fall Detection with Smart Phone Sensor". *The 3rd International Conference on Internet (ICONI) 2010*.
- Zeiler, M. D., and R. Fergus. 2013. "Visualizing and understanding convolutional networks.". *CoRR abs/1311.2901*.

AUTHOR BIOGRAPHIES

HABEN YHDEGO is a PhD student in the Modeling, Simulation and Visualization Engineering (MSVE) Department at Old Dominion University. He received MSc in Image processing and computer vision from Jean-Monet University, Saint-Etienne, France. His research interests includes applying machine learning methods for images and sensor datasets, image analysis and computer vision. His email address is hyhde001@odu.edu.

JIANG LI is an Associate professor in the the Department of Electrical and Computer Engineering at Old Dominion University. His research interests include machine learning, computer-aided medical diagnosis systems, medical signal/image processing, neural network and modeling and simulation. His email address is jli@odu.edu.

CHRISTOPHER PAOLINI is Assistant Professor in the Department of Electrical and Computer Engineering at San Diego State University. Christopher Paolini's current research interests include Internet of Things device development, machine learning, embedded systems, deep learning, high performance computing, high speed (100gbps) networking, cyberinfrastructure development, and cybersecurity. His email address is cpaolini@sdsu.edu.

MAHASWETA SARKAR is an Associate Professor in the department of Electrical and Computer Engineering at San Diego State University. Her research interest lies in the domain of wireless networks. She focuses on energy efficiency issues and "intelligence" in various applications of wireless networks. She heads the NSF funded Wireless Network Research Group and is the Education Co-director of the Center of Neuro Technology. Her email address is sarkar2@sdsu.edu.

STEVEN MORRISON is currently an Endowed Professor and Director of Research within the School of Physical Therapy and Athletic Training at Old Dominion University. A primary area of research interest is related to falls, particularly with regards to the implications changes in the pattern of gait, balance control and/or physiological function has for overall falls risk. His email address is smorriso@odu.edu.

HAMID OKHRAVI is an Associate Professor of Geriatrics at the Glennan Center, Eastern Virginia Medical School (EVMS). He is the medical director of the Memory Consultation Clinic at EVMS. He is also the medical director of Beth Shalom Village Nursing Home. His main interest in patient care and research is in the field of cognitive disorders, Alzheimer's disease and dementia. His email address is okhravhr@evms.edu.

MICHEL AUDETTE is an Associate professor of the Department of Modeling, Simulation and Visualization Engineering at Old Dominion University. He holds a PhD in Biomedical Engineering from McGill University. His research interests include medical simulation, surgery planning and medical image analysis. His email address is maudette@odu.edu.