A Comparison Between Heuristic And Machine Learning Techniques In Fall Detection Using Kinect v2

Amin Amini, Konstantinos Banitsas and John Cosmas
Department of Electronics and Computer Engineering, College of Engineering Design and Physical Sciences
Brunel University London
London, UK

***Abstract*— In this paper, two algorithms were tested on 11 healthy adults: one based on heuristic and another one on video tagging machine learning methods for automatic fall detection; both utilizing Microsoft Kinect v2. For our heuristic approach, we used skeletal data to detect falls based on a set of instructions and signal filtering methods. For the machine learning approach, we implemented a dataset utilizing the Adaptive Boosting Trigger (AdaBoostTrigger) algorithm via video tagging to enable fall detection. For each approach, each subject on average has performed six true positive and six false positive fall incidents in two different conditions: one with objects partially blocking the sensor's view and one with partial obstructed field of view. The accuracy of each approach has been compared against one another in different conditions. The result showed an average of 95.42 % accuracy in the heuristic approach and 88.33 % in machine learning technique. We conclude that heuristic approach performs more accurately for fall detection when there is a limited number of training dataset available. Nevertheless, as the gesture detection's complexity increases, the need for a machine learning technique is inevitable.**

Keywords*— Kinect; fall detection; machine learning; heuristic; AAL;

I. Introduction

Automatic fall detection is one of the most widespread research topics in healthcare and Ambient Assisted Living (AAL) as many physical conditions include falls as one of their main symptoms. Having a system that can autonomously detect a fall incident could decrease the risk of injuries and consequently the treatment expenditures. Furthermore, it helps to evaluate gait performance and fall analysis and provides valuable data for further studies.

There have been significant studies such as [1] [2] [3] [4] with regards to fall detection using different techniques over the past two decades, each with its advantages and drawbacks. Some of the earliest approaches in fall detection were based on wearable devices and attached sensors. Although accurate, they mandate the user to carry extra devices, charge batteries, wear special clothing or sensors to be attached to the body that made them uncomfortable to use. Moreover, these apparatuses may interrupt the normal daily activity and consequently gait performance analyzes. Studies that have been carried out based on wearable devices showed fairly accurate results in fall detection.

With advances in computer vision and image/video processing, newer, unobtrusive detection techniques have emerged [5] [6] [7] [8]. One of the most established technologies to eliminate the disadvantages of wearable sensors is the Kinect camera. It consists of a set of sensors that make accurate, non-invasive detection possible.

In this paper, two different techniques (heuristic and machine learning) were tested and compared, using the second edition of Microsoft Kinect sensor called Kinect v2, which is an improved version of its predecessor. The above techniques are fundamentally different and perform differently in diverse situations. Thus, a thorough comparison on the accuracy and efficiency of each technique under different conditions is needed.

II. Background

A. The Kinect v2 Sensor

Kinect is an add-on peripheral produced by Microsoft for its Xbox gaming console. It is a motion sensing apparatus that can take human natural body motions as an input. It consists of variety of sensors such as color and infrared (IR) alongside with an array of microphones. By casting IR lights on objects and calculating the time each beam takes to be received by the sensor's IR receiver, a depth map can be drawn that makes motion sensing technology possible in a 3D environment. Microsoft enabled developers to take advantage of the Kinect sensor beyond video game consoles by releasing a Software Development Kit (SDK) [9]. Since the release of Kinect SDK, developers utilized the sensor's full potential and employed it for different purposes including biomedical engineering, human-computer interaction and human posture recognition.

Kinect v2 is the second iteration of the Kinect sensor released in 2014 by Microsoft Corporation. A number of improvements has been made over its predecessor: the sensor is able to process data at two gigabits per second speed making it more accurate; its depth and infrared sensor resolution have been increased to 512x424 and its color sensor encompasses a 1080p resolution video running at 30 frame per seconds (fps) [10]. The number of skeleton joints that the sensor can detect has been increased from originally 20 to 25. Moreover, the number of concurrent user detection has also been increased from the originally two to six people. The camera's field of view has also been increased, enabling users to operate in a smaller area and closer to the sensor than before. These enhancements made Kinect v2 more accurate in data acquisition and gait performance analysis.

B. Related Works

Different techniques were used for fall detection such as the use of Kinect depth sensor [7] [8], skeleton tracking [5] and subject-to-floor distance determination. Additionally, some used a single Kinect sensor while some employed a system of multi-Kinect configuration to have a wider coverage. In our previous research [5], it was concluded that Kinect is an ideal method to detect Parkinson's disease patients' fall incidents by tracking a subject's head position and its velocity. In [6], a two-stage system was implemented consisting of an algorithm to determine the vertical state of the subject for each frame and an on-ground event trigger based on decision tree and feature extraction methods. A dataset of 454 simulated falls as well as nine naturally occurred fall were used in the trial.

In [7], a depth-based method, coupled with velocity and inactivity calculation, was used. The need for floor detection was eliminated as the method implemented a 3D bounding box for the object of interest in each frame, making the algorithm function without environmental pre-knowledge such as floor's height and position.

During another study [8], a set of raw depth data were used to extract human body features using a depth blob technique for each frame and by taking into account the position and distance of each blob from the others. Based on the implemented algorithm, a fall incident will be counted as positive if the head position is close to the ground. This was feasible because the camera was place on the ceiling facing downwards.

Finally, in [11], a series of machine learning techniques were compared for human fall detection based on different algorithms such as Native Bayes, Support Vector Machine (SVM) and decision tree, where the study showed over 93.3 % detection accuracy using decision tree method.

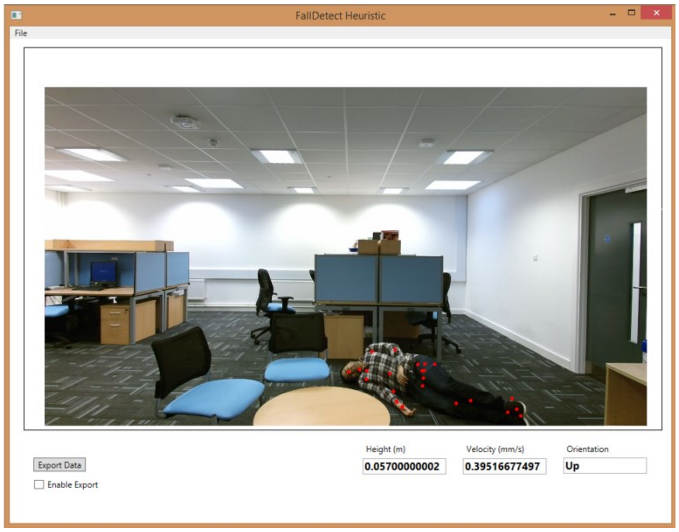
As the above studies suggested the Kinect camera as a prime candidate for vision-based fall detection systems, a comparison between different methodologies in different situation is needed to evaluate their accuracy. Moreover, only limited research [12] [13] have been conducted to assess the performance and improvements of Kinect v2 over the original Kinect. In [11], although different machine learning techniques were compared, a direct comparison between different categories of fall detection techniques is needed. Therefore, this paper compares two different categories of methods, one based on heuristic approach using skeleton tracking and one based on machine learning using video tagging technique utilizing AdaBoostTrigger algorithm.

III. Methodologies

The Project was written in C# language using Microsoft Visual Studio 2013 and Microsoft Kinect for Windows SDK 2.0.

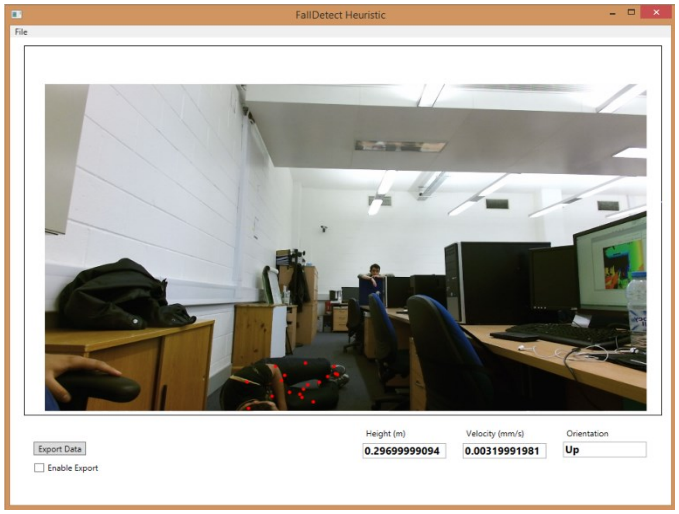
A. Heuristic Approach

For this approach, an algorithm was designed to track a subject's head 3D Cartesian coordinate location at all times. By using the Kinect skeleton tracking, the spatio-temporal position of each joint, with respect to other joints, can be determined. The proposed system holds the information of the subject head's position, velocity and acceleration for one second. This is required to calculate the average velocity and acceleration. Based on the vector that the subject's head is moving towards and the distance between the head and the floor, a fall incident can be detected if the average velocity and acceleration reach 1 m/s and the subject's head distance to ground is less than 10 cm. This minimizes the chances for false positives' occurrence by not taking into account low-velocity falls such as laying down or high ground distance incidents such as sitting on a chair. Additionally, the system is designed to distinguish between the different types of falls such as critical falls in which the subject is unable to stand up after the incident. This is achievable by adding a timer that can be user-defined to set a threshold for the maximum time elapsed before it reaches a critical falling point.



Height (m) 0.05700000002
Velocity (mm/s) 0.39516677497
Orientation Up

Fig. 1. Heuristic approach software in action (objects partially blocking the sensor's view): a subject has fallen (on the floor) and the system recorded his velocity, direction and distance to the floor.



Height (m) 0.29699999094
Velocity (mm/s) 0.00319991981
Orientation Up

Fig. 2. Heuristic approach software in action (partial obstructed field of view): a subject has fallen (on the floor) and the system recorded his velocity, direction and distance to the floor.

1) Floor Detection: A surface floor can be determined by using the scalar equation of plane.

$$\boldsymbol{Ax + By + Cz + D = 0} \tag{1}$$

Where A, B and C are the components of a normal vector that is perpendicular to any vector in a given plane and D is the height of the Kinect from the levelled floor.

Once the floor is determined, the distance of a given joint's 3D Cartesian coordinate location to the floor can be yield as follows:

$$\boldsymbol{d = \frac{Ax+By+Cz+D}{\sqrt{A^2+B^2+C^2}}} \tag{2}$$

Where x, y and z are the 3D coordinate location of a joint.

2) Acceleration and velocity: To determine the subject's head fall acceleration and velocity, Euclidean distance equation was employed to calculate the distance changes over time.

$$\boldsymbol{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}} \tag{3}$$

Where x_i, y_i, z_i and x_{i-1}, y_{i-1}, z_{i-1} are the current and past subject's head 3D Cartesian coordinates, respectively.

B. *Machine Learning Approach*

For the machine learning approach, an AdaBoostTrigger machine learning technique was implemented. It is an event detection technique that outputs a discrete or binary result. It is based on an AdaBoost machine learning algorithm that operates depending on its dataset and trainings, which combines a series of weak classifiers into a final boosted output [14]. A total of 29 minutes training videos based on 435 GB of 30 fps, 1080p uncompressed Red Green Blue (RGB) and 424p depth data were recorded and stored as a training dataset. Using Kinect Visual Gesture Builder (VGB), these videos were tagged frame by frame to specify a falling incident's true positive and false positive moments. Despite the fact that VGB software is available for developers, it only facilitates the machine learning and training process not the detection phase. Thus, a system was developed to utilize the training set produced by VGB software in order to detect the fall incidents. The information was processed to generate a series of weak and strong classifiers and calculate their confidence levels. The generated results were given to the software that was written for the machine learning approach to be compared against the real-time subject's postures. Two factors (velocity and subject's head distance to the ground) were used for the machine learning approach. The following figure demonstrates the video tagging training process.

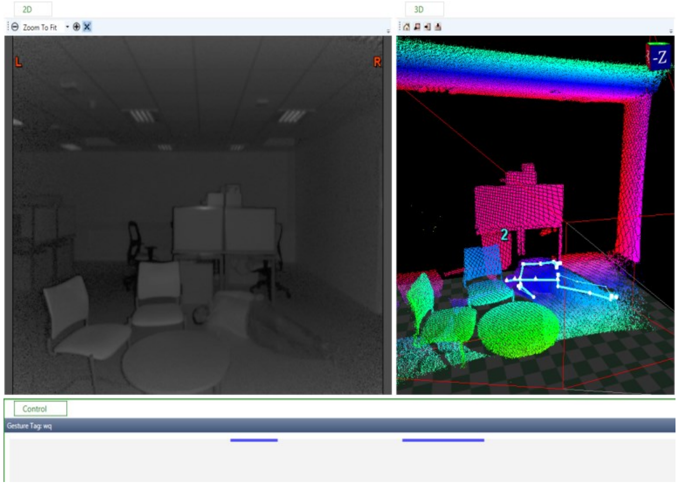
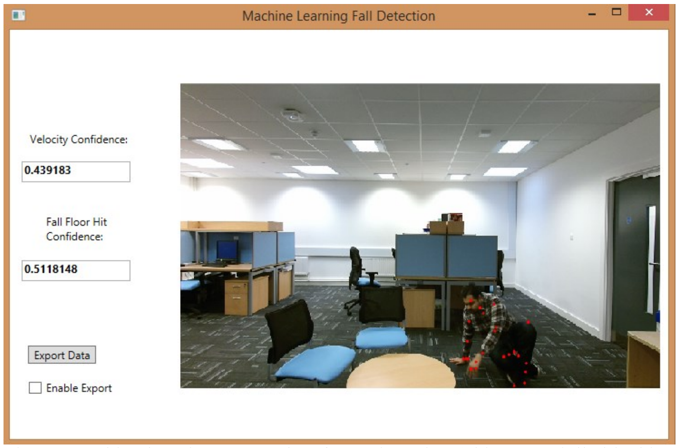


Fig. 3. Visual Gesture Builder. On the left, a true positive fall is marked and tagged for training as shown by blue bars at the bottom; on the right, colors represent the distance of 3D objects to the camera.

Fig. 4 and Fig. 5 depict the machine learning system UI in different scenarios.

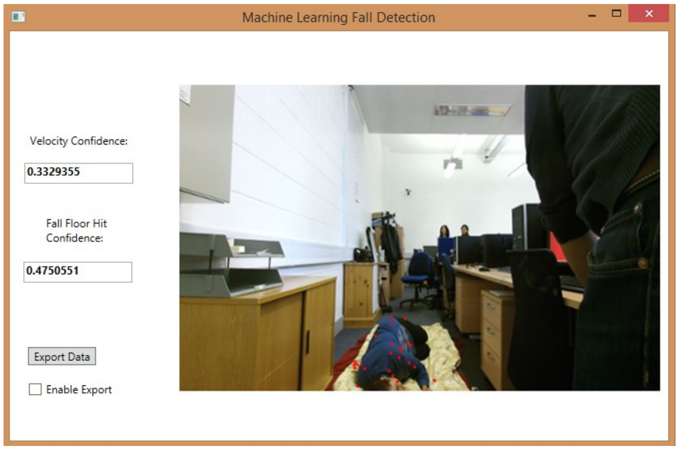


Machine Learning Fall Detection

Velocity Confidence: 0.439183

Fall Floor Hit Confidence: 0.5118148

Fig. 4. Machine learning approach software in action (objects partially blocking the sensor's view): A fall is about to happen and the system shows the confidence factors accordingly.



Machine Learning Fall Detection

Velocity Confidence: 0.3329355

Fall Floor Hit Confidence: 0.4750551

Fig. 5. Machine learning approach software in action (partial obstructed field of view): A false positive fall has happened and the system shows the confidence factors accordingly.

C. Testing Environment & Subjects

The Kinect v2 sensor was placed at a height of one meter facing parallel to the surface. Due to the Kinect v2 wider field of view, subjects were placed at a distance range of one to two and a half meters.

11 subjects (Table 1) participated in the trial for both heuristic and machine learning approach. For each approach, each subject on average performed six true positive and six false positive fall incidents. False positive incidents were performed by laying down or sitting on the floor. For machine training phase, extra postures were performed by each participant to teach the system detect false positive results.

TABLE 1. TEST SUBJECTS' CHARACTERISTICS (n=11; 8 males, 3 females)

Subject Characteristics	Mean (SD)	Range
Age	2.34	24-31
Height (cm)	8.31	163-187
Weight (kg)	16.35	51-100
BMI (kg/m²)	3.83	17.3-30.1

IV. RESULTS

For heuristic approach, as expected, the system showed good results with high accuracy. Although each subject's fall incident had different characteristics in terms of velocity and postures, the implemented algorithm detected 95.42 % of falls successfully. Fig. 6 shows a subject's head fall velocity as detected by the system.

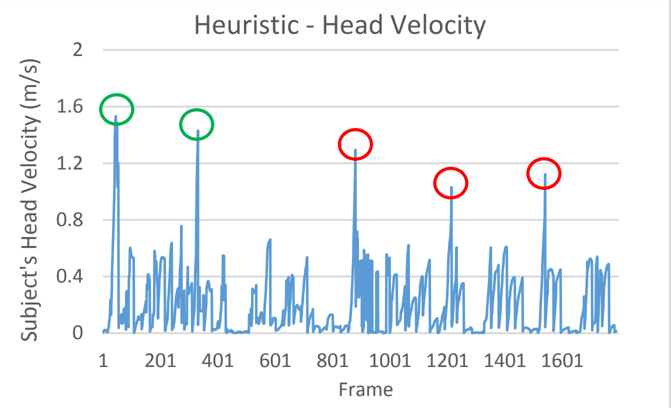


Fig. 6. Heuristic – Subject's head fall velocity (false positives and true positives are shown as red and green circles, respectively).

As the above figure shows, there are five major falls with considerable velocity detected by the system. These data then were analyzed by the algorithm and compared to the subject mean head's Y axis height (Fig. 7) to eliminate false positives. Note that the subject's height is measured as a 3D Cartesian coordinate point located in the middle of the head.

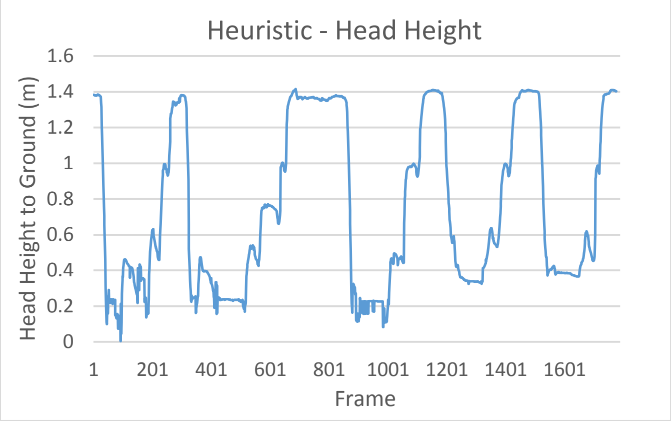


Fig. 7. Heuristic – Subject's head height to floor.

In order for the system to detect a falling incident with higher accuracy, the signal was filtered, normalized and the earlier-mentioned thresholds such as velocity, acceleration and the subject's head distance to ground were set in order to ignore false positives. A conditional statement was applied to ignore signals when the subject's head distance to the ground is higher than 10 cm or its velocity is less than 1 m/s. Fig. 8 shows the same subjects' falling incidents after correction. Note that the whole process is automatic and done in real-time by the developed system.

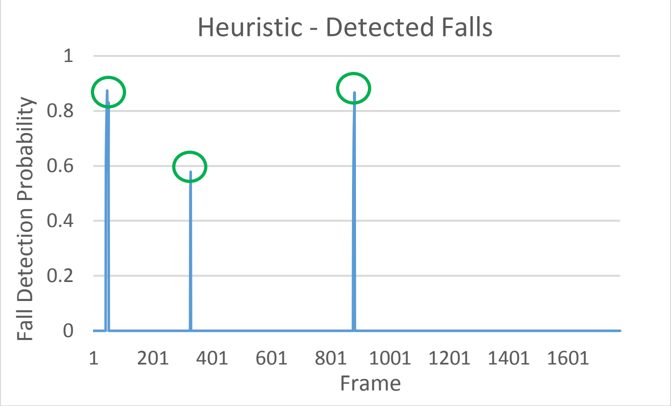


Fig. 8. Heuristic – Filtered true positive fall detection's confidence level (true positives are shown as green circles).

As the above figure illustrates, the system managed to detect three discrete fall incidents during the trial for the subject. The Y axis shows the system's confidence in fall detection with one being the absolute certainty. As the set of instructions for fall detection algorithm was implemented in software, the heuristic approach showed a similar result in both scenarios (one with objects partially blocking the sensor's view and one with partial obstructed field of view). Nonetheless, in partial obstructed field of view condition, the accuracy of true positive detection was lower depending on whether the subject's fallen body was fully seen by the Kinect. In both conditions, the obstructed joints' 3D Cartesian coordinate location tracking was compensated and predicted using 'inferred' state enumerate. By implementing the 'inferred' joint state, the joint data were calculated and its location was estimated based on other tracked joints.

For machine learning approach, two factors were taken into account. The system was built to calculate both velocity and the subject's head closeness to the ground by importing false positive, false negative and true positive tagged-video samples. Results show that a system based on a 2.4 GHz quad core PC with 8 GB of memory required about 18 minutes to calculate and process all training videos including 11 subjects' fall incidents in different conditions and 11 subjects' false positive training videos. Overall, 435 GB of 30 fps, 1080p uncompressed RGB and 424p depth video data were processed by the system for a total of 29 minutes training videos. Fig. 9 shows the likelihood of the same subject reaching the threshold fall velocity as a confidence level zero to one. False positives are shown with red circles.

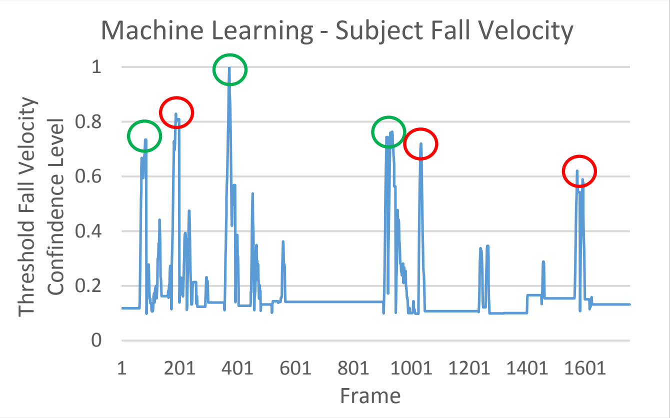


Fig. 9. Machine learning – Subject's fall velocity threshold confidence level (false positives and true positives are shown as red and green circles, respectively).

Fig. 10 shows the confidence level for detecting the same subject's distance to the ground as a fall incident happens.

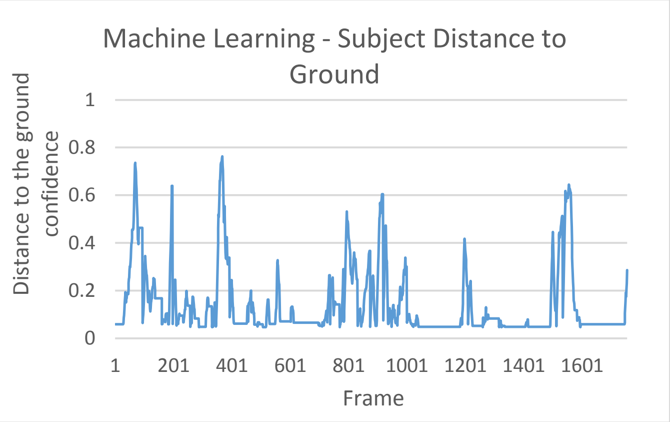


Fig. 10. Machine learning – Subject's distance to the ground confidence level.

As the above figure demonstrates, the machine learning approach proved to be less accurate compared to the heuristic method due to the limited number of subject's samples [15]. The accuracy of an AdaBoostTrigger algorithm is highly dependent on the number of training samples. Nevertheless, by introducing a second confidence factor into the equation and merging both confidence factors, the system managed to cancel out most of the false positives. Fig. 11 shows the combined confidence level for the subject's fall on the floor and fall velocity. The graph shows that once two signal are combined, most of the false positive detection was weakened and consequently, the successful detection signals have been boosted and normalized. The green circles show true positive fall incidents with highest confidence level whereas the red circle indicates an error in picking up a false positive incident as a true positive.

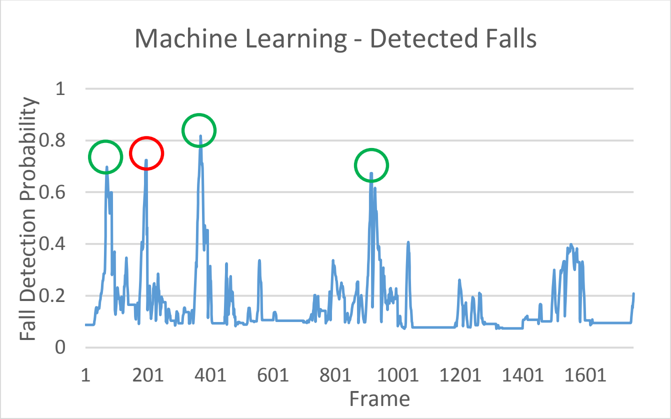


Fig. 11. Machine learning – Fall detection overall confidence from combining the threshold fall velocity and distance to the floor factors (false positives and true positives are shown as red and green circles, respectively).

Fig. 12 shows the data once the system passed it through a filter to ignore signals, which either of the probability levels (threshold fall velocity or subject's distance to ground) is below 60 %.

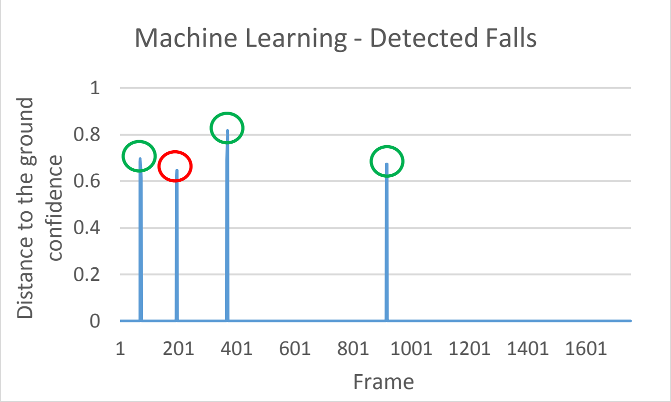


Fig. 12. Machine learning – Filtered fall detections' confidence level (false positives and true positives are shown as red and green circles, respectively).

Combining two sets of conditions yielded a slightly higher detection rate. Nevertheless, in order to observe a noticeable improvement in detection of true positives, the number of dataset and training data should be significantly increased [15].

Overall, the system behaved differently for each testing trial. The algorithm managed to detect 88.33 % of true positive falls successfully.

V. Conclusion

Two different approaches including heuristic and machine learning (using AdaBoostTrigger algorithm) fall detection were implemented, based on Microsoft Kinect v2 sensor. The efficiency and accuracy of both were compared against one another in similar conditions.

Heuristic approach showed more accuracy as it works independent to the number of pre-operation training videos. Heuristic algorithms are very efficient for discrete detections such as falls, as long as the detection case is simple enough to be implemented algorithmically. On the other hand, AdaBoostTrigger machine learning approach effectiveness is greatly dependent on the number of training samples. Correct and accurate sample tagging plays a significant role in reducing latency and increasing accuracy. Nevertheless, the overall success rate of a machine learning algorithm with a small training dataset can be increased by implementing and combining more confidence factors.

Overall, machine learning approach is ideal for detections that are more sophisticated in terms of body movements and require a lot of thresholding and variables such as complex and continues body gestures or gait disorders, but for simpler cases such as fall detection, its disadvantages outweigh its benefits; mainly due to its need for significant amount of system resources (i.e. CPU and memory) to process information beforehand. Moreover, video tagging is a painstaking task and requiring a lot of time and training data.

Thus, we conclude that for fall detection with a small number of training samples (11), the heuristic approach provides results that are more accurate. Nonetheless, by increasing the number of training data, the accuracy of the machine learning algorithm would also be increased. Machine learning approach accuracy would be significantly higher in complex scenarios where a continuous and sophisticated gesture needs to be detected.

A. Future Work

Our research team plans on investigating the effectiveness of a machine learning approach based on a combination of both AdaBoostTrigger and Random Forest Regression Progress (RFRProgress) when Freezing of Gait (FOG) symptoms in Parkinson's disease patients occurs. This will be compared against a traditional heuristic approach and their accuracy and success rate would be evaluated.

References

[1] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensor,* vol. 15, no. 5, pp. 4544-4553, 2015.

[2] J. Min Kang, T. Yoo, H. Chan Kim, "A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare," *IEEE Transactions on Instrumentation and Measurement,* vol. 55, no. 5, pp. 1655-1661, 2006.

[3] M.N Nyan, Francis E.H. Tay, E. Murugasu, "A wearable system for pre-impact fall detection," *Journal of Biomechanics,* vol. 41, no. 16, pp. 3475-3481, 2008.

[4] T.T. Nquyen, MC. Cho, TS. Lee, "Automatic fall detection using wearable biomedical signal measurement terminal," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Minneapolis, MN, 2009.

[5] A. Amini Maghsoud Bigy, K. Banitsas, A. Badii, J. Cosmas, "Recognition of postures and fog in Parkinson's disease patients using Microsoft Kinect sensor," in *7th International IEEE/EMBS Conference on Neural Engineering (NER)*, Montpellier, France, 2015.

[6] E.E. Stone, M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," *IEEE Journal of Biomedical and Health Information,* vol. 19, no. 1, pp. 290-301, 2015.

[7] G. Mastorakis, D. Makris, "Fall detection system using Kinect's infrared sensor," *Journal of Real-Time Image Processing,* vol. 9, no. 4, pp. 635-646, 2014.

[8] G. Samuele, C. Enea, S. Susanna, G. Ennio, "A depth-based fall detection system using a Kinect® sensor," *IEEE Sensors,* vol. 14, no. 2, pp. 2756-2775, 2014.

[9] M. N. Center, "Microsoft Releases Kinect for WindowsSDK beta for academics and enthusiasts," Microsoft, 16 June 2011. [Online]. Available: http://www.microsoft.com/en-us/news/press/2011/jun11/06-16MSKinectSDKPR.aspx. [Accessed 18 June 2015].

[10] C. Amon, F. Fuhrmann, F. Graf, "Evaluation of the spatial resolution accuracy of the face tracking system for Kinect for windows v1 and v2," in *6th Congress of Alps-Adria Acoustics Association*, Graz, Austria, 2014.

[11] M. Strémy, A. Peterková, "Comparison of machine learning methods for the purpose of human fall detection," *Research Papers Faculty of Materials Science and Technology Slovak University of Technology,* vol. 22, no. 35, pp. 69-76, 2014.

[12] B.F. Mentiplay, L.G. Perraton, K.J. Bower, Y.H. Pua, R. McGaw, S. Heywood, R.A. Clark, "Gait assessment using the Microsoft Xbox One Kinect: Concurrent validity and inter-day reliability of spatiotemporal and kinematic variables," *Journal of Biomechanics,* vol. 48, no. 10, pp. 2166-2170, 2015.

[13] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, A. El Saddik, "Evaluating and improving the depth accuracy of Kinect for Windows v2," *Journal of Biomechanics,* vol. 48, no. 10, pp. 2166-2170, 2015.

[14] Kankanala, P. ; Das, S. ; Pahwa, A. , "AdaBoost ^: An Ensemble Learning Approach for Estimating Weather-Related Outages in Distribution Systems," *IEEE Transactions on Power Systems,* vol. 29, no. 1, pp. 359 - 367 , 2014.

[15] C. Marais, "Gesture Detection Using Machine Learning," 14 November 2011. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=28066. [Accessed 12 October 2015].