2022 IEEE International Conference on Multisensor Fusion and Integration (MFI 2022),
20-22 September 2022, Cranfield, UK
DOI:10.1109/MFI55806.2022.9913877

Vision-based Fall Detection in Aircraft Maintenance Environment with Pose Estimation

Adeyemi Osigbesan, Solene Barrat, Harkeerat Singh, Dongzi Xia, Siddharth Singh, Yang Xing, *IEEE Member*, Weisi Guo, *IEEE Member,* and Antonios Tsourdos

***Abstract*— Fall-related injuries at the workplace account for a fair percentage of the global accident at work claims according to Health and Safety Executive (HSE). With a significant percentage of these being fatal, industrial and maintenance workshops have great potential for injuries that can be associated with slips, trips, and other types of falls, owing to their characteristic fast-paced workspaces. Typically, the short turnaround time expected for aircraft undergoing maintenance increases the risk of workers falling, and thus makes a good case for the study of more contemporary methods for the detection of work-related falls in the aircraft maintenance environment. Advanced development in human pose estimation using computer vision technology has made it possible to automate real-time detection and classification of human actions by analyzing body part motion and position relative to time. This paper attempts to combine the analysis of body silhouette bounding box with body joint position estimation to detect and categorize in real-time, human motion captured in continuous video feeds into a fall or a non-fall event. We proposed a standard wide-angle camera, installed at a diagonal ceiling position in an aircraft hangar for our visual data input, and a three-dimensional convolutional neural network with Long Short-Term Memory (LSTM) layers using a technique we referred to as Region Key point (Reg-Key) repartitioning for visual pose estimation and fall detection.**

I. Introduction

Maintenance environments can be hazardous, with dangerous examples such as unattended machinery running, lack of improper fencing/physical guards near hazardous locations, and cluttered workspaces. In Great Britain alone, for the past few years, 25-30% of fatalities were related to maintenance activities [1], the one of the most fatal accidents for workers is falling from a height [1]. Falls are defined as the event of an individual unintentionally approaching the rest position on the floor or other lower level, leading to injuries that are fatal or non- fatal. They are the second leading cause of unintentional injury deaths worldwide with an estimated 684,000 fatal injuries occurring each year [2].

Given the high number of fatal falling injuries occurring each year, those in maintenance environments which are non-fatal, if detected and reported promptly, reduces the chances of further injury or death, thus this work proposes an inte- grated computer vision based system to provide monitoring and pose estimation of employees whilst working in an aircraft maintenance environment in addition to the already in place control measures such as harnesses, guardrails and scaffolding [1]. The output of the system intends to benefit management staff or other employees working in a loud environment, alerting them to a hazardous fall for quick intervention.

In this study, a fall detection system was developed based on the work done by Fan et al. [6]. The system can be divided into four parts, the first is the video input from continuous camera feed, the second is the human pose extraction, the third part is the human pose classification, and finally, the output or classification result interpretation. The main contribution of this study are as follows. 1) Human posture detection using body joints position estimation over a time span with a technique we termed Reg-Key repartitioning and 2) Fall prediction using a CNN-LSTM Model with an accuracy of 80.5%. This study is organized as follows, section II detail out a review of existing literature in the realm of fall detection. Section III describes the methodology used to define the system. Section IV highlights the design details from the human posture extraction to the data processing approach, as well as details of the model training and testing. In section V, the results of the CNN-LSTM model are presented, and a discussion of outcome was done in VI while we highlighted possible future work in section VII before the study conclusion in section VIII.

II. Literature Review

In spite of the scope of this project being fall detection in aircraft maintenance environment, we have extensively researched documented studies around the topic of fall detection in various other settings. Typically, approaches based upon machine learning (ML), Internet of things (IoT), and imaging techniques are in common use. These methods can be classified into three categories; vision/camera-based methods, wearable methods, and ambience methods [3]. The most relevant of these methods with respect to this paper would be solutions based on ML technology that could be synchronized with wearable technologies through sensor fusion.

A. Camera-Based Methods

This methodology utilises a non-intrusive fusion of video footage and motion analysis with computer vision to detect falls. To begin, the work done by Bian et al. [4] demonstrates an approach to fall detection by tracking joints of the human body using a single depth camera. Joints were tracked using a randomized pose-invariant decision tree algorithm to extract joints with a support vector machine (SVM) classifier. Detecting fall using 3D head trajectory analysis. By mounting

*Research supported by Centre for Autonomous and CPS, Cranfield University.

A. Osigbesan, S. Barrat, H. Singh, D. Xia, and S. Singh are with the School of Aerospace, Transport, and Manufacturing, Cranfield University. Email:a.o.osigbesan.147, solene.barrat.436, harkeerat.singh, dongzi.xia.486, siddharth.singh.026@cranfield.ac.uk

Y. Xing, W. Guo and A.Tsourdosare are with the Centre for Autonomous and Cyber-Physical Systesm, Cranfield University. Email: yang.x, weisi.guo, a.tsourdos@cranfield.ac.uk.

a depth camera close to the ceiling, occlusions were minimized, and they achieved an accuracy score of 97.9% with a low error of 2.1% by just tracking the head for their proposed motion analysis. The system struggled to detect falls if the subject falls on higher platforms like furniture.

Another CB method that uses SVM is the work done by Yu et al. [5], [5] proposed a background subtraction technique to extract the human body silhouettes, describing posture with ellipse fitting and using unsupervised one class SVM to find abnormality in daily posture. Although their system falls short of being autonomous, requiring human intervention for the segmentation and selection of video clips, they achieved 100% true fall detection performance with just 3% false detection.

Similar to the work of Yu et al. [5], Fan et al. [6] presented a vision-based fall detection approach with analysis of the human posture extracted from image sequences of moving body parts. The posture analysis process was done in three stages; 1) Body Extraction using a colour distortion model by Horpraset et al. [7], 2) Human Posture Description using Normalized Directional Histogram (NDH) and a statistical hypothesis testing to differentiate standing, lying, crouching, and sitting postures, 3)Using NDH data, a Directed Acyclic Graph Support Vector Machine (DAGSVM) and a majority voting method to differentiate a fall from a non-fall event over a temporal time window, a detection of up to 95.2% accuracy was achieved on a public fall dataset.

In more recent study, the work of Huynh-The et al. [16] leverages the high dimensional data capability of deep convolutional neural network (DCNN) to learn human action by extracting features from a 3D skeleton data using depth camera. Joint-to-joint distance and orientation within and between consecutive frames were encoded into color pixel action images. [16] used transfer learning to finetune a pre-trained Inception-v3 network model, achieving up to 90.33% accuracy on the most challenging NTU RGB+D dataset [17].

Overall, camera-based systems can be useful and advantageous given their environment agnostic qualities as regards to installation as well as their capability to leverage improving ML algorithms with edge computing. They however exhibit some issues related to occlusions, lighting, and field of view coverage, thus, our proposal in this paper will be to utilize more advance deep learning approach, such as the use of CNN for image processing, posture analysis and action classification.

B. Wearable-Sensor-Based Methods

The growth in micro-electro-mechanical systems (MEMS) led to sensor technology becoming more compact and low cost. Their integration into available alarm systems or into accessories carried by a subject became a feasible and efficient way of non-intrusive and non-invasive diagnosis and monitoring. Xu, et al. [12] and Perry et al. [13], similar to Kwolek and Kepski [11], discussed how the use of accelerometer wearable sensors leads to more accurate fall detections systems as compared to other non-accelerometric systems along with their low power and cost advantages. They identified a few disadvantages as the sensitivity to environmental conditions such as gravity, inability to create contextual understanding of output data leading to high false positives, privacy concerns etc.

Wu, et al. [14] carried out a study that concludes that adding a Gyroscopic Sensor to an already present accelerometer system for fall detection improved accuracy as the gyroscope takes angular velocity and object orientation into account. With advancements in MEMS applications of health sensors as electromyogram (EMG) sensors and cardiotachometer, also used with accelerometers have also been used for fall detections by monitoring muscle control signals, change in heart rate etc. [15].

In summary, wearable sensors can be efficient for fall detection when used with other present sensor systems (usually combining accelerometer/gyroscopes) due to their lack of contextual data output and the possibility of unrecognized noise in data. These systems tend to use simpler threshold-based classifiers such as decison trees, RandomForest Classifier, K-means clustering etc. [18].

III. METHODOLOGY

Bearing in mind our original objective, i.e., to design an intelligent computer vision-based system for fall detection within aircraft maintenance environment, we scoped our study to focus on a single-actor fall detection in an aircraft hangar. We are proposing the use of wide-angle cameras installed at diagonal ceiling position within the hangar for our visual data input, while our fall detection model will adopt a CNN-LSTM model which allows for batch training and time distribution understanding of the input data.

Although we understand that the total number of cameras needed is environment specific, we will suggest a minimum of 3 cameras for reasonable handling of occlusion. This num- ber may however be increased based on the unique settings of different hangars. Video input from the camera system is divided into frames which gets annotated for the subject's keypoints and effectively serves as input to the CNN-LSTM model classifier algorithm. The system architecture described in Fig. 1 is as follows:

1) Process the Input video and extract pose features of subject in individual frames using pose estimation.

2) Compute Reg-Key and discriminate a state of fall/falling event from normal/non-fall event on ex- tracted frame.

3) Output and Display results if an individual has fallen over.

The output has been designed with providing visual indica- tion of the fall, as well as a notification to the environment to ensure that the fallen over individual is given attention quickly. Falls are determined by trained AI modules, which determines whether an individual is fallen over based on their Region Keypoint (Reg-Key) matrix. More detailed breakdown of the individual processes is covered in the preceding sections of this paper.

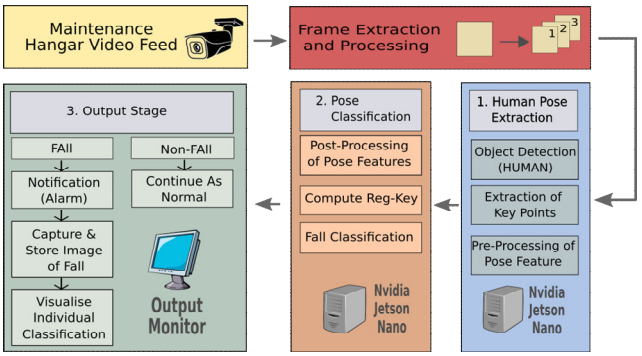

Fig. 1. Proposed Fall Detection System Architecture

IV. Experiment Design

A. Pose Estimation

Highlighting human locations for a frame in computer vision can be done via methods such as Object Detection and Pose Estimation, using pre-trained models widely available online. Object Detection is essentially identifying objects of predetermined classes within a set of digital pixels array and presenting such identified objects in annotated squares called bounding boxes [21]. The annotated bounding box describes the size and position of the object in a two-dimensional plane relative to the image frame. Pose estimation on the other hand uses a set of key points that are equivalent to joint positions on a human, the intent is to have a skeleton-based overlay attached to each person within any given frame.

Pose estimation could use any of two separate techniques, Top-Down and Bottom-Up [19] [20]. Top-Down approach works by using an object detector to determine the human position, then estimating the key points within the bounding box. Bottom-Up approach detects each joint/body part in the scene and constructs the skeleton joint frame for each person. To generate human joint position estimation described above, we used a pre-trained pose estimator model called MoveNet. This allowed us to detect and generate 17 key point on a moving or stationary human body captured in video frames seamlessly and accurately.

1) **MoveNet Keypoint Detection**: MoveNet is a bottom-up pose estimation model, that is used in our system to obtain the keypoints with the MoveNet Lightning model, given its high-performance in latency-critical applications. The model utilises a feature extractor called MobileNetV2 attached with a feature pyramid network (FPN) for higher resolution, rich feature map outputs (Tensorflow) [8].

2) **Region Keypoint Repartitioning**: This is the second stage in the pose estimation process. The body keypoints obtained as output from MoveNet is not used in its raw form as input to the neural network, but used to create what we termed as Region Keypoints (Reg-Key) in a process called Region Keypoint repartitioning. It is a way to summarise the posture of a human into a simple matrix. The principle is to divide the human body into 8 fixed regions starting from the centre of the hips, then each region is described as a matrix that stores information about the keypoints within it. This method is inspired by the work done by Fan et al. [6]. The process can be broken down into these three steps as seen in Fig. 2:

1) Extract the position of hips of the detected human and compute the mean position between them to define the region center.

2) Using the mid-hip point, separate the regions using, one vertical, one horizontal and two 45-degree diagonal lines. From these lines, a polygon is computed to englobe the delimited surface. The polygon will go from the center of the hips to the boundary of the frame and the regions are defined from R1 to R8. If the hips are higher or lower than the middle (x and y coordinates) of the frame, then there are cases to consider.

3) Each key point is verified in which polygon region it is contained in and stored into a (Number of Regions × Number of Key points) matrix.

As this matrix describes the overall posture, we can differentiate if someone is walking or falling using extraordinarily little computational memory. This can help us avoid feeding the neural network with heavy input images and lowers the risk of a slow system. We could attempt to further minimize input size by summing up all the columns of the Reg-Key matrix into a simpler vector, but this led to low accuracy for the trained model as it substantially reduces information about the posture, thus we used the Reg-Key matrix in its original size.

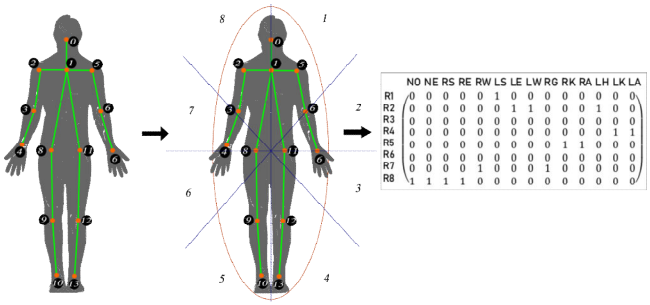

Fig. 2. Reg-Key Process Visualized.

B. Dataset

With an understanding that a typical aircraft hangar is a large open space with very high vertical room space and three or less full walls, usually well-spaced to accommodate the aircraft horizontal stretch [11], we were biased to source for, and the use existing dataset captured using wide-eyed cameras placed at high edges to cover a very broad view of the space under surveillance.

The apparent absence of ready-made airport hangar maintenance dataset clearly presents a possibility of a bias towards a more generalized fall detection that may not adequately account for maintenance specific scenario and potential for occlusion. This necessitated the need to generate dataset by experimentation in a real airport hangar.

The Cranfield University maintenance hangar at the DARTeC building was used for the experimentation and data collection exercise after adequate planning and risk assessment was carried out. Video recording was captured simultaneously from three camera positions with the subject recreating regular maintenance actions that included safe short distance falls and trips at various locations outside and inside of the aircraft. To ensure our training set can handle occlusion,

a few of the scenes captured featured part of the aircraft body obstructing the camera view of the subject. All simulations were carried out under careful supervision of the hangar's safety manager to ensure participants follow all safety guidelines as detailed in the risk assessment document. Apart from the standard safety gear, the subject also used addition safety gear, like elbow and knee pads, winter jackets for harsh weather situations and an air mattress to dampen simulated falls.

About 50 short (2-5 minutes) videos of simulated maintenance activities were recorded, some with falls and others without. The captured videos were stripped into frames and annotated using MoveNet pose estimator and the Reg- Key matrix describing different postures were generated as described in the Section IV.A.

Fig. 3. Cranfield Local Data Collection.

C. Data Processing

With an homogeneous mix of sourced fall scenario dataset and the data generated from experimentation, each scene for each camera angle is separated into smaller videos for every action, and labelled for the fall output.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop-down menu to differentiate the head from the text.

The cut videos were injected into the MoveNet/Reg- Key repartition program that was developed using Python programming language. This program detects the key points of every human in the video with MoveNet and computes the Reg-Key matrix for all the frames automatically. If multiple humans are detected, they are labelled with a track ID in the Reg-Key matrix. Some of the time, there are errors, when the program detects non-existent humans on the frames, therefore, it became necessary to sort the output matrix, deleting the few wrong detections. We then validate our repartition results visually with the output video using MoveNet.

The Reg-Key data needs some processing to have a structured input that keeps regularity through actions and time. The data was reorganized using Pandas data manipulation library for Python [10] to achieve a consistent input for our model training. The index of the created Reg-Key data frame was used to regroup the Reg-Key matrix for every scene, camera, and action. At the end, we had a total of 36,011 frames with exploitable fluid human detection.

D. Time-Series Generator

In order to be sure that the training data will be accepted by the neural network we pre-process it into a Time Series Generator (TSG). It is useful for Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) to maintain state across samples within a batch. This generator will structure the data (input and output) into samples ready to train the supervised deep learning model. The issue is that TSG is limited to one-step outputs, therefore, it was necessary to compute TSG on each cut videos that were stored into the Reg-Key data frame, and then concatenate them together with the input and output separately.

D. Machine Learning Algorithm Design

The Reg-Key repartition that we are injecting into our neural network is comparable to an image: it has two dimensions, and it describes a spatial and physical depiction. For this reason, a CNN is adapted to Reg-Key analysis and posture feature extraction. Keeping in mind that our problem has time dependency, our goal here is to interface a CNN with Long Short-Term Memory (LSTM) to take advantage of their strengths. A CNN layer helps extract feature specificities of spatial environments while LSTM supports sequence prediction. Their combination is perfectly suited for video classification. In other words, LSTM layer needs to build internal state across the sequence of image interpretations that has been handled by the CNN layers. A Time Distributed layer englobing the CNN layers will convert the output of the CNN into a sequence that the LSTM can process. The global architecture of CNN-LSTM model is structured as follows.

Fig. 4. High-Level CNN-LSTM Model Architecture

The choice of number of nodes *N* for the LSTM layer can be determined from the following formula:

$$N = \frac{N_s}{\alpha \times (N_i + N_o)} \quad (1)$$

with *Ns* being the number of samples in training set, *Ni*, the number of input neurons, *No*, the number of output neurons and α, an arbitrary scaling factor between 2 and 10. If the number of nodes of the LSTM layer stays inferior to this value, overfitting has better chances to be avoided.

The final reference model used to perform video classification is a 3D CNN layer. The 3D Conv layers preserve a link through time as it outputs a temporal volume. The number of filters defines how many volumetric outputs we will have. We used inputs of size ($n \times 8 \times 14 \times 1$) where n is the time series frames of 8×14 matrices on one channel. We kept the batches of time frames as our input data and the output remains a binary vector of size n. Fig. 5 visually describes the model architecture and Table IV shows a comparison of all models architectures that was tried.

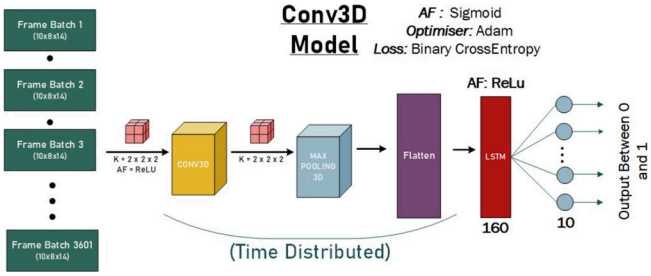

Fig. 5. Convolutional 3D Model with LSTM Layers and Sigmoid activation function.

F. Model Training

The Model described above is Optimized using Adam Optimizer, given its reliable performance, the loss function (Binary CrossEntropy) and activation function (Sigmoid) are chosen to limit the output values between 0 and 1. At the end of the training epochs, all the models, 1D, 2D and 3D achieved approximately the same loss value, with the third model being slightly better.

G. Model Testing

Once the three models were trained using our training data, it was important that the model was evaluated on its performance to accurately detect falls. The models were evaluated using a standard confusion matrix to obtain the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), to generate 4 evaluation metrics of Accuracy, Error Rate, Sensitivity and Specificity. F1-score was also used to check for the harmonic mean between the Sensitivity and Specificity.

$$Accuracy\,(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Error\ Rate(Er) = \frac{FP + FN}{TP + TN + FP + FN} \tag{3}$$

$$Sensitivity/Recall\,(Se) = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity\,(Sp) = \frac{TN}{TN + FP} \tag{5}$$

$$F1\ Score = 2 \times \frac{Sp \times Se}{Sp + Se} \tag{6}$$

As the confusion matrices are a visualisation of the ground truth labels against the model predictions, it is possible to conclude that the model is capable of predicting falls. The data obtained from these matrices can be used further to evaluate the test data assessed upon our model. Looking at Fig. 6 and Table III, a few conclusions can be drawn. First, the model had zero FP classifications which suggests the model does not misclassify a fall. This was one of our primary concerns (triggering false alarm). Second, there are 940 True negatives for each Model, this could likely be due to each test data containing a portion of non-falls (classified as 0) before the actor falls over. Finally, there are quite a few FNs where the system believes there is not a fall. This could be very constraining for our system. It could mean that the model leaves out some falls and would not fulfil its purpose. Some additional statistics are needed to validate the true performance of the model.

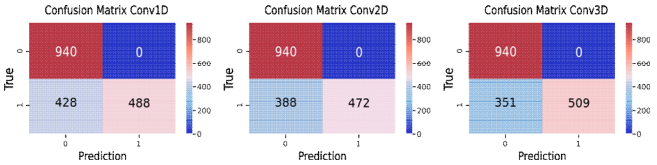

Fig. 6. Confusion Matrix Plots for the Conv1D, Conv2D and Conv3D Models.

Table I
CONFUSION MATRIX SUMMARY

Model	True Pos	False Pos	False Neg	True Neg
Conv1D	488	0	428	940
Conv2D	472	0	388	940
Conv3D	509	0	351	940

Table II
CONFUSION MATRIX SUMMARY

Model	Accuracy (%)	Error (%)	Sensitivity (%)	Specificity (%)	F1-Score
Conv1D	76.9	23.1	53.3	100	69.5
Conv2D	78.4	21.6	54.9	100	70.9
Conv3D	80.5	19.5	59.9	100	74.9

V. Results

From the summarized metric in Table IV, obtained using the equations 2 through 6, the accuracy improves from 76.9% for the first model, to 80.5% for the final model, this reinforces the reasoning behind improving the model from Conv1D to Conv3D. The specificity is likely 100% due to the model being able to classify all non-fall movements con- ducted in the test dataset, but an issue is the low sensitivity score. As the F1 score is quite high, 69.5% for Conv1D, 70.9% for Conv2D, and 74.9% for Conv3D, it once again confirms the model's robustness on finding falls. Looking at the metrics once again, it is possible to finally conclude quantitatively that the third model is the best proposed model for detecting falls in our system.

After the testing prediction process, there were obviously a few errors in the output: some 0 were lost within the sequences of 1s characterizing a fall. To avoid those unnecessary variations, we used a common technique popular in video classification. For every prediction we performed an average on the previous 10 frames.

VI. Discussion and Future Works

Essentially, our 3D CNN-LSTM model presented a very stable and consistent fall prediction, standing out in the area of accuracy and speed of prediction. The general observation is that the 1D CNN-LSTM is slower to predict the fall. We can easily observe that there is no time link between the frames. It waits for the body to be fallen and to be horizontal before predicting a fall. It is not sensitive to the loss of balance that occurs prior to a fall. The 2D CNN- LSTM is faster than the first model but it also lacks rapidity in identifying a fall. It is difficult to conclude that the model has a better sense of time. But if we place our judgement on the statistics, we can agree that the more the complexity of the model increases and the more dimensions we use, the more the model is accurate.

We can assume that the 3D model is better because it has the better statistics. We need to visualize the results on a video where the fall prediction is displayed to be sure of the consistency of our results. It is also valid to remark that the

predictions of our models can be subjective and biased under our judgement because of manual labelling.

Although reasonable result was achieved in this study, we cannot discount the fact that availability of more data will significantly improve the training and validation processes, therefore, in the future, more data could be generated for model training, validation and testing purposes, especially from an aircraft maintenance environment. While doing that, it is important to ensure that the subject simulating a fall spends more time in a fallen state, as this creates a more realistic scenario for the model training.

To further deal with cases of occlusion, strong consideration must be given to the possibility of interfacing the camera-based approach discussed in this study with wearable technologies using sensor fusion. We would like to propose a creative combination of accelerometer and gyroscope sensor data fusion with the pose description features extracted from the Reg-Key repartitioning. Another improvement that will benefit this study in future is the real-time synchronisation of the video feed and the fall detection algorithms as this study utilized only recorded video footage.

VII. Conclusion

In this work, we propose a working fall detection system by using a 3D CNN-LSTM model. We used a new method that we called Reg-Key repartition; it provides a matrix containing condensed information of the human posture through keypoint estimation via MoveNet. Three models were tried in this work, that have been iteratively ameliorated to reach the Conv3D model with an accuracy, sensitivity, and specificity of 80.5%, 59.9% and 100%, respectively. The visual output of the testing data is more than satisfying. We notice a real implication of the time link between frames that the other models did not have. The model could be trained and tested with k-folds in order to truly validate it. Given the simplicity of the models, we believe that it can integrate easily into embedded systems so long as the prerequisites are compatible with the embedded hardware.

References

[1] Health and Safety Executive, ”Hazards during maintenance.” hse.gov.uk. https://www.hse.gov.uk/safemaintenance/hazards.htm (accessed June 1, 2022)

[2] World Health Organisation, ”Falls” Fact Sheet; 26 April 2021, who.int.https://www.who.int/en/news-room/fact-%20sheets/detail/falls (accessed June 1, 2022)

[3] R. Tanwar, N. Nandal, M. Zamani and A. Abdul Manaf ”Pathway of Trends and Technologies in Fall Detection: A Systematic Review” Healthcare 2022 (MDPI), p. 3. https://doi.org/10.3390/healthcare10010172

[4] Z. Bian, J. Hou, L. Chau and N. Magnenat-Thalmann ”Fall Detection Based on Body Part Tracking Using a Depth Camera” IEEE Journal of Biomedical and Health Informatics, Vol 19, No.2 March 2015.

[5] M. Yu, Y. Yu, A. Rhuma, S. M. R., L. Wang and J.A. Chambers ”An Online One Class Support Vector Machine-Based Person-Specific Fall Detection System for Monitoring an Elderly Individual in a Room Environment” IEEE Journal of Biomedical and Health Informatics, Vol 17, No. 6, pp. 1002-1014.

[6] K. Fan, P. Wang, Y. Hu and B. Dou ”Fall detection via human posture representation and support vector machine” International Journal of Distributed Sensor Networks 2017, Vol 13(5).

[7] Horprasert T, Harwood D and Davis L. ”A statistical approach for real-time robust background subtraction and shadow detection” In: Proceedings of the IEEE con- ference computer vision, Kerkira, 20?27 September 1999, pp.1-19. New York: IEEE.

[8] Tensorflow Blog, ”Next-Generation Pose Detection with MoveNetand TensorFlow.js”blog.tensorflow.org. https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with- movenet-and-tensorflowjs.html.

[9] E.M. Rantanen, T.A. Butts, D.S. Wojtowicz, and M.L. Webb ”Human Factors Considerations in the Design of an Aircraft Maintenance Hangar” Proceedings of the Human Factors and Egonomics Society 47th Annual Meeting-2003, p. 930.

[10] Pandas, ”Pandas Documentation.” pandas.pydata.org. https://pandas.pydata.org/docs/ (accessed May 1, 2022).

[11] B. Kwolek and M. Kepski ”Improving fall detection by the use of depth sensor and accelerometer” Neurocomputing Vol 168, 30 November 2015, pp. 637-645.

[12] T. Xu, Y. Zhou, and J. Zhu ”New Advances and Challenges of Fall Detection Systems: A Survey” Applied Sciences 2018, 8(3), p. 418.

[13] J.T. Perry, S.Kellog, Sundar M. Vaidya, Jong-Hoon Youn, Hesham Ali and Hamid Sharif ”Survey and evaluation of real-time fall detection approaches”ieeexplore.ieee.org https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5423081.

[14] W.Wu, S. Dasgupta, Ernesto E Ramirez, Carlyn Peterson, and Gregory J Norman ”Classification Accuracies of Physical Activities Using Smartphone Motion Sensors” https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3510774/.

[15] S. Yang and S. Lin ”Fall detection for multiple pedestrians using depth image processing technique”Computer Methods and Programs in Biomedicine Vol 114, Issue 2, April 2014, pp. 172-182.

[16] T. Huynh-The, C. Hua, T. Ngo and D. Kim ”Image representa- tion of pose-transition feature for 3D skeleton-based action recognition” Information Sciences 513 (2020) 112-126.

[17] A. Shahroudy, J. Liu, T. Ng and G. Wang ”NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1010-1019.

[18] P. Vallabh and R. Malekian. ”Fall detection monitoring systems: a comprehensive review” Journal of Ambient Intelligence and Humanized Computing (2018) 9:1809-1833 https://doi.org/10.1007/s12652-017-0592-3

[19] Z. Cao, T. Simon, S. Wei and Y. Sheikh ”Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields” Computer Vision and Pattern Recognition, arXiv:1812.08008 https://doi.org/10.48550/arXiv.1812.08008.

[20] L. Song, G. Yu, J. Yuan and Z. Liu ”Human pose estimation and its application to action recognition: A survey” Journal of Visual Communication and Image Representation Volume 76, April 2021, 103055.

[21] Mostafa S.Ibrahim, Amr A.Badr, Mostafa R.Abdallah and Ibrahim F.Eissa ”Bounding Box Object Localization Based On Image Superpixelization”Procedia Computer Science Vol. 13, 2012, pp. 108-119.

Vision-based fall detection in aircraft maintenance environment with pose estimation

Osigbesan, Adeyemi

2022-10-13

Osigbesan A, Barrat S, Singh H, et al., (2022) Vision-based fall detection in aircraft maintenance environment with pose estimation. In: 2022 IEEE International Conference on Multisensor Fusion and Integration (MFI 2022), 20-22 September 2022, Cranfield, UK
https://doi.org/10.1109/MFI55806.2022.9913877