ELDERLY HEALTH MONITORING SYSTEM WITH FALL DETECTION USING MULTI-FEATURE BASED PERSON TRACKING

Dhananjay Kumar[1]*, Aswin Kumar Ravikumar*[1]*, Vivekanandan Dharmalingam*[1]*, Ved P. Kafle*[2]

[1] Department of Information Technology, Anna University, MIT Campus, Chennai, India
[2] National Institute of Information and Communications Technology, Nukui-Kitamachi, Koganei, Tokyo, Japan

ABSTRACT

The need for personalized surveillance systems for elderly health care has risen drastically. However, recent methods involving the usage of wearable devices for activity monitoring offer limited solutions. To address this issue, we have proposed a system that incorporates a vision-based deep learning solution for elderly surveillance. This system primarily consists of a novel multi-feature-based person tracker (MFPT), supported by an efficient vision-based person fall detector (VPFD). The MFPT encompasses a combination of appearance and motion similarity in order to perform effective target association for object tracking. The similarity computations are carried out through Siamese convolutional neural networks (CNNs) and long-short term memory (LSTM). The VPFD employs histogram-of-oriented-gradients (HoGs) for feature extraction, followed by the LSTM network for fall classification. The cloud-based storage and retrieval of objects is employed allowing the two models to work in a distributed manner. The proposed system meets the objectives of ITU Focus Group on AI for Health (FG-AI4H) under the category, "falls among the elderly". The system also complies with ITU-T F.743.1 standard, and it has been evaluated over benchmarked object tracking and fall detection datasets. The evaluation results show that our system achieves the tracking precision of 94.67% and the accuracy of 98.01% in fall detection, making it practical for health care system use. The HoG feature-based LSTM model is a promising item to be standardized in ITU for fall detection in elderly healthcare management under the requirements and service description provided by ITU-T F.743.1.

Keywords – CNN, fall detection, HoG, LSTM, object tracking

1. INTRODUCTION

According to the United Nations (UN) report on ageing world population [1], the population of elderly people will rise to 2 billion by the year 2050. The guidelines for Integrated Care for Older People (ICOPE) [2] released by the World Health Organization (WHO) clearly indicates that accidental fall is one of the common reasons for the decline in the health of the elderly. The surveillance system for effectively monitoring the elderly's health can be achieved by either a sensor or vision-based system, or a combination of both. The sensor-oriented surveillance systems generally utilize accelerometer and GPS sensors to locate the person [3]. Although, these sensors provide highly accurate real-world coordinates, there exists a possibility of sensors being misplaced, not worn by the user, or worn by the wrong user, thus restricting the tracking ability of the system. Although other alternative methods employing devices like thermal sensors have been proposed to work, they work only within a short range [4-5]. On the whole, sensor-based tracking techniques heavily rely on the assumption that users continuously wear the devices. In general, sensor-based fall identification involves the use of triaxial accelerometer sensor [6] which records real-world 3D coordinates. The continuous analysis of coordinate information also poses difficulty in differentiating between daily activities such as sleeping, sitting and standing, implying the need of more sophisticated and accurate systems based on artificial intelligence techniques such as deep learning. Vision-based fall detection using optical flow and convolutional neural networks (CNNs) [7] can be used to extract temporal features needed for improving system performance. However, existing customized techniques such as curvelets [8] do not extract deep features for human representation to detect falls.

Visual object tracking can be categorized into two broad categories namely detection-based tracking and detection-free tracking. The detection-based tracking consists of three main components: moving object detection, object classification and localization, and object tracking [9]. The moving object detection component identifies the salient objects that are present in the current frame using bounding boxes. Object classification is carried out in identifying the detected objects and segregating into specific classes, while the tracking is performed for target association in subsequent frames. On the other hand, detection-free tracking does not involve recognition of different objects, rather it utilizes motion features in order to locate moving objects. Typical detection-free tracking involves the usage of optical flow, background subtraction in order to eliminate static objects in each frame. Tracking based on background subtraction in video usually requires manual intervention to identify scene-specific objects [10]. An optical flow-based tracking algorithm also requires additional support from appearance modelling in order to produce accurate results. Optical flow along with blob analysis yields better traffic surveillance systems [11]. However, these algorithms are only capable of tracking generic objects rather than specific objects, which may lead to a reduction in the efficiency of the system.

Detection-based tracking algorithms first identify the target object to be tracked and then find the object in each frame of the video. Unique object identification can be achieved with the help of salient features the object possesses. Many methods have exploited the object appearance as an important feature to represent it in a numerical way. The appearance features such as histogram of oriented gradients, scale invariant feature transform and local Binary Pattern have significantly improved the overall accuracy of object detection and tracking.

Some of the difficulties of existing feature extraction methods have been overcome by CNN in more complex object segmentation-cum-detection processes [12-13]. Further extension of the generalized object detection using CNN [14] allows specific object tracking to be performed with high precision. The usage of Long Short-Term Memory (LSTM) helps in inferring deeper features from time-series data, thus posing it as a potential technique to be coupled with CNN for multiple object tracking. Siamese CNN helps in finding similarities in consecutive frames, due to its identical sub-network components [15].

The ITU-T Focus Group on Artificial Intelligence for Health (FG-AI4H) has considered “Falls among the elderly” [16] as one of the key areas that needs to be addressed for better healthcare. Although curvelet coefficient-based fall detection techniques [7] have translation and scaling invariant properties, detection accuracy suffers in complex background and moving objects. A machine learning-based approach [8] can handle complex scenarios of detection, but training a CNN-based generic network is not only inefficient, but also difficult to achieve a higher accuracy of fall detection in real-time environments.

To address the above limitations, we propose a system that utilizes machine-learning techniques to improve its performance accuracy. The major contribution of this work is twofold: a person tracker that considers both appearance and motion features for target association, and a fall detector that considers the sequence of person orientations. Our models are designed to leverage deep-learning techniques while complying with the criteria set by the ITU FG-AI4H. Both the models have been developed as per Recommendation ITU-T F.743.1 – “Requirements for intelligent visual surveillance”. In our system, target recognition and association are achieved with the combination of CNN and LSTM to uniquely distinguish persons from other objects. The core of the system is HoG for feature extraction which is an LSTM based model for fall detection, a promising candidate for standardization in ITU.

The remainder of the paper is organized as follows. Section 2 provides a background overview and section 3 describes the proposed system. The implementation detail for performance evaluation and experimental results are presented in section 4, while section 5 concludes the paper.

2. BACKGROUND

2.1 HOG Feature Descriptor

HoG-based feature extraction [17] uses edge orientations for object detection. It operates on grayscale image and its workflow is as follows: Initially gradient computation is carried out for each pixel in the image, by placing a mask on the image with a pixel as its center and performing element-wise multiplication. The orientations of these gradients are further found out and a histogram of orientations is created for each block. This is then subjected to both local and global normalization to finally produce the required feature descriptor.

2.2 CNN Based Feature Extraction

CNN [12] utilizes kernels for automatic deep feature extraction, classification or detection. The CNN model commonly consists of two important segments: automatic feature extraction and dimensionality reduction. Feature extraction is achieved with the help of convolutional layers. A convolutional layer consists of different kernels which learn different features, through backpropagation. The stacking of different convolutional layers allows learning of deeper features. Dimensionality reduction is achieved with the help of pooling layers and dense layers. A combination of all these layers allows the construction of a CNN model that can be utilized to solve different domain problems.

2.3 Long Short-Term Memory Network

A long short-term memory (LSTM) network [18] is a recurrent neural network that performs well for time-series based analysis in extracting temporal features. The LSTM network is made of stacks of cells in order to represent the sequential data better. An LSTM cell consists of an input gate, an output gate, and a forget gate. The input gate allows new information to enter into the cell, while the forget gate helps in remembering only the important information regarding the input data in achieving higher performance. The LSTM cell incorporates a sigmoid activation function to restrict the information flow within it and *tanh* function in order to remember relevant features.

3. PROPOSED SYSTEM

The architecture of the proposed system, as shown in Figure 1, consists of three components: client, server and cloud service. The overall workflow in the proposed system is as follows. A client, typically a hospital room or elderly home, is configured to stream videos, to the receiver (medical center/care takers) over HTTP using ITU-T H.264 encoding. The frame processing and video analysis are carried out on the server end where both MFPD and VPFD models are executed to track persons and detect occurrences of human falls. The detected person’s location and image is stored in the cloud along with the fall occurrence status. An alert is generated at the concerned client end, either a hospital or a

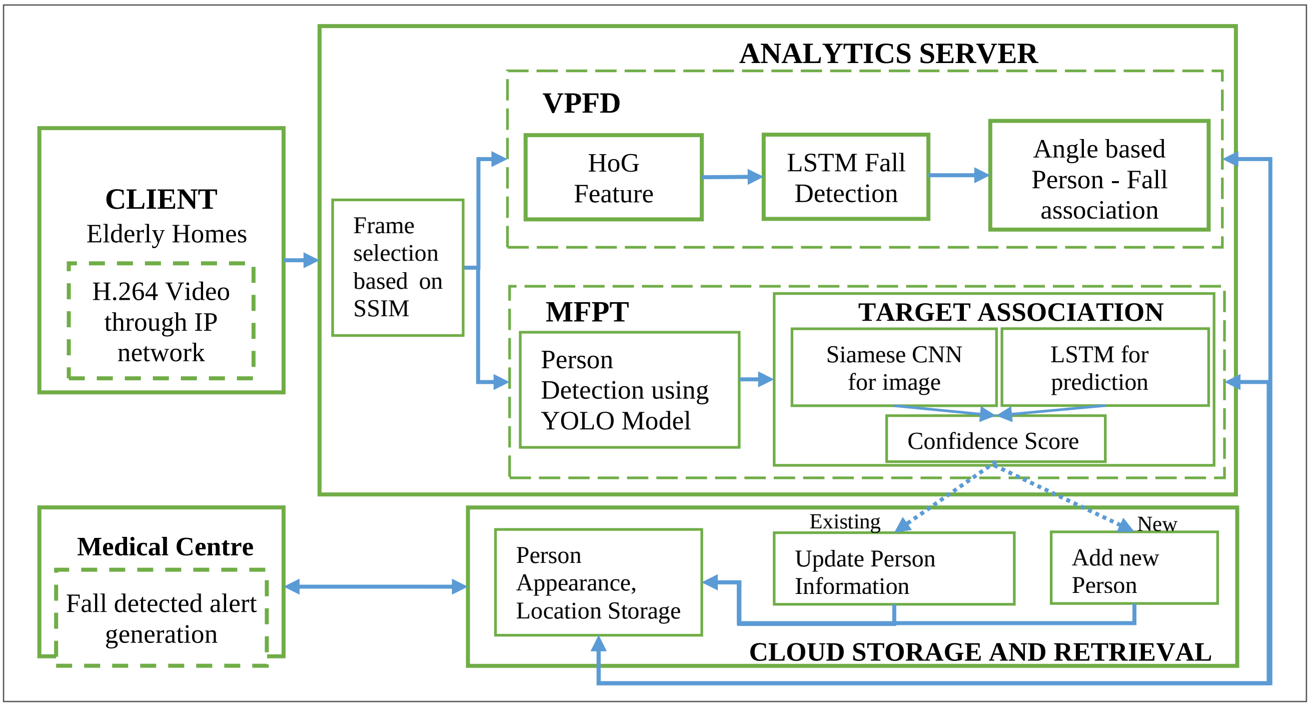
ANALYTICS SERVER
VPFD
HoG Feature
LSTM Fall Detection
Angle based Person - Fall association
CLIENT
Elderly Homes
H.264 Video through IP network
Frame selection based on SSIM
MFPT
Person Detection using YOLO Model
TARGET ASSOCIATION
Siamese CNN for image
LSTM for prediction
Confidence Score
Medical Centre
Fall detected alert generation
Existing
New
Person Appearance, Location Storage
Update Person Information
Add new Person
CLOUD STORAGE AND RETRIEVAL

Figure 1 – The architecture of the proposed model

family member, to provide location information of the persons who has fallen.

3.1 Key Frame Selection

In the server end, key frame selection has been carried out to improve the overall processing speed of the system without degrading its performance. This is achieved by comparing the previous processed frame and incoming frame using the structural similarity index (SSIM), which is calculated by Equation 1.

$$f(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (1)$$

Where μ_x, σ_x^2 are the mean and variance of pixels in image x respectively and μ_y and σ_y^2 are the mean and variance of pixels in image y, respectively.

The SSIM index value is subjected to a custom threshold to process only dissimilar images by the system. Similar images are simply skipped for faster video processing.

3.2 Person Detection

The decoded frames from the preprocessing stage are given to the object detector to accurately classify and localize different objects present in each frame. This is achieved with the help of the CNN-based YOLO model [19]. The given frame is fed as input to the YOLO model, which divides it into segments and finds objects in each segment, along with their bounding box coordinates and confidence score. The confidence score has been used as a threshold to eliminate false positive detections. The YOLO model trained on the COCO dataset [20] has the ability to detect many different object classes. The output of the model is then filtered to contain bounding box values of class ‘person’ only.

3.3 Target Association

Target association is the process of mapping the existing objects with newly detected objects from the current frame. After preprocessing and person detection stages, one of two possibilities are tested, either any of the previously moving persons could have moved to the new position or a new person could have started moving. Using this fact, tracking can be carried out for all persons, who enter and exit the scene in the video. This architecture performs tracking of the detected persons from the CNN using two distinct features, namely the visual feature and motion feature. The visual feature denotes the image similarity that helps find whether the currently found person matches with the appearance of one of the existing persons. The motion feature denotes the possibility of the existing person moving from his/her previous location to the location of the currently detected object. Visual and motion features are obtained using Siamese CNN and LSTM respectively. The utilization of dual features allows the handling of sudden entry and exit of persons in the given video.

3.4 Image Similarity

Siamese CNN, shown in Figure 2, is a neural network model that operates on a pair of images and produces a score denoting the appearance similarity between the two images. Bounding box coordinates obtained in the previous step have been used to extract the person's image from the frame. The previously stored person's images are then compared with the extracted image. The feature extraction layers in this CNN network are made the same for both images, thus making it vertically symmetrical. The resultant features of both images are merged by finding the element-wise squared difference. It is then fed into fully-connected layers for dimensionality reduction and finally to obtain the similarity score.

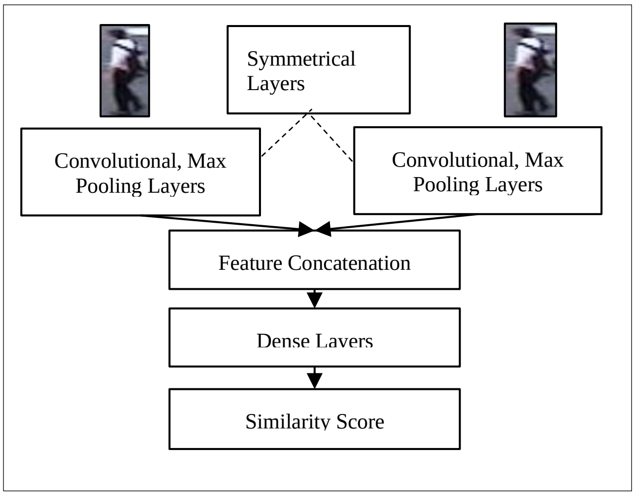

Figure 2 - Image similarity using Siamese CNN

A custom threshold is employed, where scores greater than a threshold value are considered similar and vice versa (Algorithm 1).

```
Algorithm 1: Image similarity
------------------------------------
Input:  $P_{prev}$ – Previously detected persons
        $P_{curr}$ – Currently detected persons
Output: $S$ – appearance similarity matrix
$Model_{app}$ = Trained Siamese CNN model
$M$ = count ($P_{curr}$)
$N$ = count ($P_{prev}$)
$S$ = Null matrix of dimensions $M*N$
for $i$ = {0,1, …., M-1}
        $img_{curr} = P_{curr}[i].img$
        for $j$ = {0, 1, ……, N-1}
                $img_{prev} = P_{prev}[j].img$
                $input = (img_{curr}, img_{prev})$
                $score = Model_{app}(input)$
                if $score > threshold$:
                     $S[i][j] = score$
                end if
        end for
end for
return $S$
```

3.5 Motion Similarity

Motion prediction has been used in the proposed system in order to associate objects based on their recent movements. This has been implemented using LSTM on the basis of the previous 12 center coordinates of stored persons. This vector is fed as input to the Motion LSTM model which performs temporal processing and predicts the new center for each stored person. This center indicates the next possible position of the person. The predicted centers are compared with the centers of the currently detected persons via Euclidean distance and inverse of this value is considered as the overall motion score (Algorithm 2).

```
Algorithm 2: Motion similarity
------------------------------------
Input:  $P_{prev}$ – Previously detected person
        $P_{curr}$ – Currently detected person
Output: $S$ – motion similarity matrix
$Model_{motion}$= Trained motion LSTM model
$M$ = count ($P_{curr}$)
$N$ = count ($P_{prev}$)
$S$ = Null matrix of dimensions $M*N$
for $i$ = {0,1, …., M-1}
    $center_{curr} = P_{curr}[i].center$
      for $j$ = {0, 1, ……, N-1}
          $center\_seq_{prev} = P_{prev}[j]$.center [{1, 2…,12}]
             $center_{pred} = Model_{motion}$ ()
             $dist$ = Euclid_Dist ($center_{curr}$, $center_{pred}$)
             $score = 1/dist$
             if $score > threshold$:
                   $S[i][j] = score$
             end if
        end for
end for
return $S$
```

3.6 Object Mapping

The mapping of previous to current persons is achieved by finding the best candidate for each currently detected person from the appearance and motion similarity matrices. A map data structure helps in store the detected persons in each video frame efficiently. The map stores information in distinct key-value pairs. A unique ID is assigned for every person appearing at any part of the video. Each person detected in the frame is stored in the map data structure with an ID as the key and the bounding box coordinates, frame number in which he/she was detected and the previous center list as value. During the retrieval of the person for target association, only the candidates which have a frame number less than the current frame number are considered, in order to avoid mismatching. The best candidate for the current person is selected by choosing the person with the highest appearance similarity and with a motion similarity greater than a specific threshold. The best candidate values are updated to match the current target. If no such match is found for the target, then the target is newly added into the map data structure. This data structure is stored in the cloud and both MFPD and VPFD access the data. This reduces the overall workload and allows multiple computers to run tasks in parallel improving the processing speed. The map data structure is made persistent in order to last till the end of the video instead of removing the objects that have temporarily exited, which helps in handling short-term occlusions. Also, this persistent data structure allows persons to be tracked through different cameras in the surveillance system.

3.7 Fall Detection

The VPFD model constitutes two important phases: feature extraction using HoG and sequence analysis using LSTM. For each frame, the image is converted to grayscale and resized to a dimension of 640 x 480 in order to maintain the same dimension of the resultant feature vector during the training and testing phase. The HoG feature vector is then computed from the reshaped image and passed on to the LSTM model. For each frame, the HoG features of the previous three frames, including the current frame are considered as input to the LSTM model. Temporal sequence of feature vectors is taken into account to eliminate false detection, thus improving the capability of differentiating falls and actions of daily-living. The model output indicates the occurrence of a fall. In order to map the fall with the detected person, angles of previous center coordinates are calculated as shown in Equation 2. The midpoint of the lower boundary line of the image is taken as the reference plane for angle computation as shown in Figure 3.

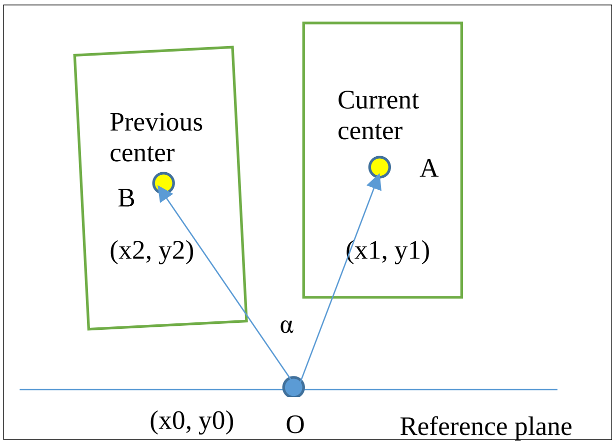

Figure 3 – Angle between two centers of same person

Vector-based notation is utilized to represent both OA and OB vectors respectively, as given in following equations.

$$OA = (\mathrm{x1} - \mathrm{x0})i + (y1 - y0)j \qquad (2)$$

$$OB = (\mathrm{x2} - \mathrm{x0})i + (y2 - y0)j \qquad (3)$$

The angle between vectors OA and OB can be obtained by finding the cosine inverse of the dot product of two vectors as:

$$\cos\alpha = \frac{OA\,.\,OB}{|OA||OB|} \qquad (4)$$

The person with an angle greater than the specific threshold is determined as the fallen person (Algorithm 3). The possibility of multiple persons falling at the same time can also be taken into consideration in this approach. The threshold value helps in eliminating false fall detection instances.

```
Algorithm 3: Fall Detection
-----------------------------------
Input: curr – current frame index
           seq – sequence of frames
           $P_{curr}$ – Currently detected persons
Output: FP – Fallen person
$Model_{fall}$ = Trained HoG-LSTM model
N = count ($P_{curr}$)
T = time window
max_angle = 0
FP = None
        data = empty array
        for j = {0,1, ……, T-1}
            feature = HoG(seq[i-j])
            data.append(feature)
        end for
        val = $Model_{fall}$ (data)
        if val > threshold
            for k = {1,2, ……., N}
            angle = find_avg_angle ($P_{curr}$[k].center, T)
                if max_angle < angle
                    max_angle = angle
                    FP = k
                end if
            end for
        end if
return FP
```

4. IMPLEMENTATION AND RESULTS

The proposed system has been implemented and tested on the Intel i5 processor CPU and Nvidia GeForce 940MX GPU over standard datasets. The MFPT model is trained on the Object Tracking Benchmark (OTB - 100) dataset and Multiple Object Tracking (MOT) dataset. The UR Fall dataset [21] is utilized to train the VPFD model. The proposed system has been implemented using the Python programming language. FFMPEG has been utilized for video encoding and decoding purposes. The image pre-processing techniques are executed with the help of OpenCV library. The Keras library in Python was used to create both the CNN and LSTM deep-learning models. The overall performance of the proposed system is shown in Figure 4, where a red boundary indicates that a person is falling.

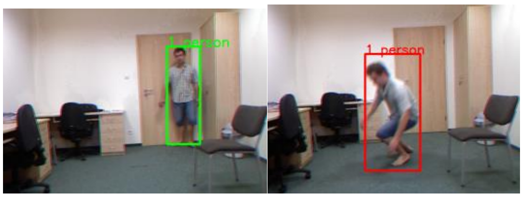

Figure 4 – System applied on UR Fall dataset

4.1 Siamese CNN Implementation

The Siamese CNN has been trained on a custom dataset generated from OTB and MOT datasets. This custom dataset is created by extracting the images of all the persons present in all the videos of the dataset using ground truth information. Similar and non-similar pairs of images are generated from the custom dataset by pairing images of the same object and pairing images of different objects respectively. The training parameters of this network are shown in Table 1. The network is trained such that similar pair inputs produce a score closer to 1 and dissimilar pair inputs produce a score closer to 0.

Table 1 – Siamese CNN parameters

S. No.	Parameter	Value
1	Learning Rate	0.001
2	Optimizer	Adam
3	Total epoch	5
4	Train image split	70%
5	Test image split	30%
6	No. of convolutional layers used	9
7	No. of Pooling layers used	4
8	No. of Dense layers used	2
9	Threshold for image similarity	0.5

The trained model is then subjected to network pruning in order to increase the processing speed of the model. The overall training and validation phase of the model, shown in Table 2, indicates the maximum accuracy attained after 5 epochs.

Table 2 – Validation phase of Siamese CNN

Epoch	Loss	Accuracy (%)
1	0.3879	82.22
2	0.2716	86.39
3	0.2014	87.82
4	0.1862	91.78
5	0.1713	92.07

4.2 LSTM Implementation

A custom dataset containing the center coordinates of objects from OTB and MOT datasets has been utilized for the overall training of the motion-LSTM model. From the dataset, random sequences of length 12 are extracted as inputs to the model and centers of each sequence are considered as the actual outputs. The model is then trained with this data sequence using the parameters shown in Table 3.

The training phase results in Figure 5 show the model convergence with respect to the input data after 350 epochs. The slight increase of loss during the 150th epoch indicates that the model slightly falls in the local minimum rather than attaining global minimum. The use of Adam optimizer helps to regularize the parameters during this stage and allows to further decrease the overall loss value of the network.

Table 3 –Motion LSTM parameters

S. No.	Parameter	Value
1	Learning Rate	0.001
2	Optimizer	Adam
3	Total epoch	350
4	Train split	80%
5	Test split	20%
6	No. of LSTM units used	25
7	Euclidean distance threshold for motion similarity	10

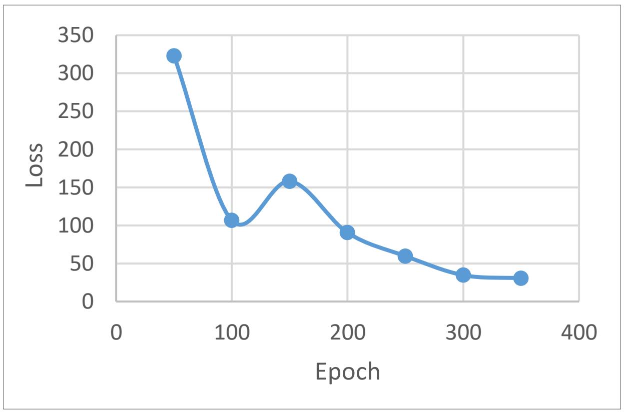

Figure 5 – Validation phase of LSTM

4.3 MFPT Results

The performance of MFPT has been analyzed using the following two metrics: precision and multiple object tracking accuracy (MOTA). Precision is the measure of detecting objects with appropriate bounding boxes. This is calculated by finding the Manhattan distance between the predicted bounding box center and actual bounding box center. If the distance is less than the threshold then it indicates correct object detection. The average precision of the tracker for all the person videos in the OTB 100 dataset is shown in Figure 6. The percentage of average precision at threshold value of 20 is 94.67%.

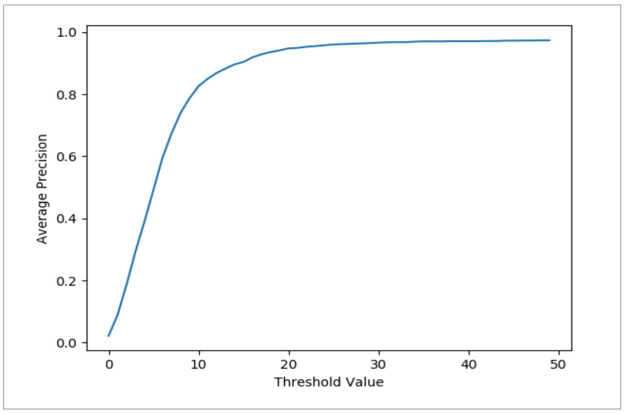

Figure 6 – Average precision in OTB 100 data set

MOTA, denotes how well the tracker is able to map the person to a unique ID from the entrance till the exit of the object from the video. This metric is calculated with the help of four parameters, namely the number of correct detections, number of misses, number of wrong detections and number of ID switches. The correct detections denote the assignment of correct IDs to corresponding persons. Misses denote the count of persons that the tracker did not detect. Wrong detections signify the action of the tracker to make false person detections and the ID switches denote the number of times the object's ID has been changed. The overall MOT accuracy is calculated using Equation 5.

$$MOTA \;=\; 1 - (M + WD + ID_{switch}) \,/\, (Obj_{gt}) \qquad (5)$$

Where *M* denotes person misses, *WD* denotes wrong person detections, ID_{switch} represents ID switches and Obj_{gt} denotes total persons in the entire video.

The accuracy for the MOT dataset along with the four mentioned parameters is listed in Table 4. This table also shows the performance comparison of three different sub components. The results show that the combination of appearance and motion similarity yields higher accuracy.

Table 4 – MOTA results

Method	Correct Detects	Miss	Wrong Detects	ID switch	MOTA
CNN + LSTM	78.23%	12.2%	3.3%	7.5%	76.6%
CNN	77.1%	15.4%	7.01%	7.5%	70.1%
LSTM	78.96%	14%	8.1%	7.1%	70.8%

4.4 VPFD Results

The UR Fall dataset has been utilized for the training and validation phase of the VPFD model. The fall dataset consists of 30 Fall event videos and 40 normal videos containing daily life activities. The ground truth specifies whether fall has occurred in each and every frame of the videos. Every frame in the dataset video is resized to 640 x 480 in order to maintain uniformity in feature dimension. After applying HoG, the training sequences for LSTM are generated by considering three consecutive frames and their feature vectors. The output for this sequence is majority voting of the ground truth values for each frame. These sequences are passed as training input to the LSTM model initialized with parameters as shown in Table 5.

The validation phase of the fall detector in Table 6, indicates that the VPFD model has learnt to differentiate between fall and non-fall sequences with high accuracy.

The accuracy comparison of various methods in Table 7 show that better feature extraction and effective time series representation can improve the overall performance of the fall detector.

Table 5 – Fall LSTM parameters

S. No.	Parameter	Value
1	Learning Rate	0.001
2	Optimizer	Adam
3	Total epoch	6
4	Train split	80%
5	Test split	20%
6	No. of LSTM units used	64

Table 6 – Validation phase of VPFD

Epoch	Loss	Accuracy %
1	0.2937	87.42
2	0.1401	93.45
3	0.1051	96.52
4	0.0874	97.68
5	0.1211	95.20
6	0.0553	98.01

Table 7 – Comparison of methods based on accuracy

S. No.	Method	Accuracy %
1	Curvelets + HMM [7]	96.88
2	Optical Flow + CNN [8]	95.00
3	HoG + LSTM (Proposed)	98.01

Although fall detection methods based on curvelets and HMM [7] produce higher accuracy than the optical flow technique with CNN [8], the proposed technique employing HoG features in LSTM achieves significantly higher accuracy due to an enhanced learning technique.

5. CONCLUSION

The proposed system is based on the combination of two models: MFPT and VPFD to monitor an elderly person's health related activities and report any falls detected through video surveillance. The system is designed to work within a confined location such as hospitals, indoor rooms and public places. The system has been tested on two different datasets of MOT and UR Fall and evaluated the performance of both models. The MFPT model's precision and accuracy denote the fact that multiple feature-based models help in achieving higher efficiency. The proposed system achieved 94.67% precision in tracking and 98.01% accuracy in elderly fall detection. The usage of LSTM model in both the models has aided in representing time-series data effectively. The proposed system for elderly healthcare in homes and hospitals can be standardized in ITU-T Study Group 16, which is the parent group of Focus Group on Artificial Intelligence for Health (FG-AI4H). The proposed work can be extended to detect different activities apart from fall detection, and recognize and report in the cases of anomalies. The fall detection module consisting of a HoG feature-based LSTM training network is the standardization item.

REFERENCES

[1] World Population Aging Report, United Nations 2017, https://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2017_Highlights.pdf

[2] WHO Guidelines on Integrated Care for Older People: https://www.who.int/ageing/publications/guidelines-icope/en/

[3] S. Van der Spek, J. Van Schaick, P. De Bois, and R. De Haan, "Sensing human activity: GPS tracking," Sensors vol. 9 no. 4, pp. 3033-3055, 2009.

[4] S. K. Opoku, "An indoor tracking system based on bluetooth technology," arXiv preprint arXiv 1209.3053, 2012. https://arxiv.org/ftp/arxiv/papers/1209/1209.3053.pdf

[5] M. Shankar, J. B. Burchett, Q. Hao, B. D. Guenther, and D. J. Brady, "Human-tracking systems using pyroelectric infrared detectors," Optical Engineering, vol 45, no. 10, 2006.

[6] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," International Journal of Telemedicine and Applications, Vol 2015 Article ID 576364, 2015

[7] N. Zerrouki, and A. Houacine, "Combined curvelets and hidden Markov models for human fall detection", Multimedia Tools and Applications vol 77 no. 5 pp. 6405-6424, 2018.

[8] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras, "Vision-based fall detection with convolutional neural networks", Wireless Communications and Mobile computing, Volume 2017, Article ID 9474806, 2017.

[9] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods, " International Journal of Innovative Research in Computer and Communication Engineering vol 2, no. 2 pp. 2970-2979, 2014.

[10] A. Aggarwal, S. Biswas, S. Singh, S. Sural, and A. K. Majumdar, "Object tracking using background subtraction and motion estimation in MPEG videos," Asian Conference on Computer Vision, Springer, pp. 121-130, 2006.

[11] S. Aslani,, and H. M. Nasab, "Optical flow based moving object detection and tracking for traffic surveillance," International Journal of Electrical, Electronics, Communication, Energy Science and Engineering" vol. 7, no. 9 pp. 789-793, 2013.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[14] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," European Conference on Computer Vision, pp. 749-765., 2016.

[15] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu and Y. Zheng, "Siamese Convolutional Neural Networks for Remote Sensing Scene Classification," IEEE Geoscience and Remote Sensing Letters. vol. 16, no. 8, pp. 1200-1204, Aug. 2019.

[16] ITU focus group on "Artificial Intelligence for Health" https://www.itu.int/en/ITU-T/focusgroups/ai4h/Pages/default.aspx

[17] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Conference on Computer Vision & Pattern Recognition (CVPR'05), vol. 1, pp. 886-893. IEEE Computer Society, 2005.

[18] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural Computation vol. 9, no. 8 pp. 1735-1780, 1997.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Procs. of the IEEE Conferece on Computer Vision and Pattern Recognition, pp. 779-788. 2016.

[20] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," European Conference on Computer Vision, pp. 740-755, 2014.

[21] B. Kwolek, and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," Computer Methods and Programs in Biomedicine vol. 117, no. 3 pp. 489-501, 2014.