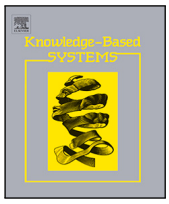
Knowledge-Based Systems 311 (2025) 113038

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Real-time fall detection on roads using transfer learning-based granulated Bi-LSTM

Anima Pramanik [a,d], Soumick Sarker [b], Sobhan Sarkar [c,*], Sankar K. Pal [d]

[a] Information Systems, Indian Institute of Management Ahmedabad, Vastrapur, Ahmedabad 380015, India
[b] Department of Chemical Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, West Bengal, India
[c] Information Systems & Business Analytics, IIM Ranchi, Ranchi, Jharkhand 835303, India
[d] Indian Statistical Institute, Center for Soft Computing Research, 203 Barrackpore Trunk Road, Kolkata 700 108, India

ARTICLE INFO

Keywords:
Pedestrian safety
Fall
Granulation
Transfer learning
Bi-LSTM

ABSTRACT

In this study, a new deep learning-based model, namely, transfer learning-based granulated Bi-LSTM (TLG-LSTM) is developed for fall detection on roads. The TLG-LSTM can handle the uncertainty issue arising between various 'Fall' and 'No Fall' events in complex scenarios concerning both indoor (i.e., home) and outdoor (i.e., road) areas. The TLG-LSTM consists of four phases: (i) object detection and tracking, (ii) MoveNet-Lightening and object-level feature(s) computation for all the detected objects, (iii) granule formation using these features, and (iv) temporal self attention mechanism-based Bi-LSTM for granule classification as 'Fall' or 'No Fall'. Unlike state-of-the-art models, TLG-LSTM uses both MoveNet-Lightning and object-level features, enabling better modeling of both indoor and outdoor falls. For each detected object, two MoveNet-Lightning features, namely head-hip distance and hip-ankle distance are defined and used for obtaining a granule, namely pose granule. Whereas, three object-level features, namely change in aspect ratio, speed variation, and change in area are used for obtaining another granule, namely object granule. The commonality between these two granules represents the approximate regions concerning fall scenarios. Instead of the entire frame, these common granules are fed to the Bi-LSTM network for fall classification, thereby increasing speed as well as accuracy. Moreover, temporal self attention mechanism-based transfer learning is used to re-train the Bi-LSTM network, enhances the training speed and accuracy. Characteristics of TLG-LSTM are demonstrated over several real-time traffic videos acquired from 'YouTube8M'. The superiority of the developed TLG-LSTM is also claimed over several state-of-the-art models.

1. Introduction

According to the statistics of World Health Organization (WHO) [1], fall on road is the second leading cause of unintentional traffic injury death worldwide. Annually, approximated 684,000 individuals succumb to death due to falls. Most of the fallen people are senior citizen. As per WHO statistics [1], annually, approximated 37.3 million cases including both fatal and non-fatal falls require medical attention. Falls can have devastating repercussions for people of all ages, but the elderly are particularly vulnerable [2]. On road, due to various traffic events (e.g., collision, near-miss, etc.), 'Fall' may occur. As a consequence of 'Fall', pedestrians may get injured. Sometimes, 'Fall' may occur on road due to either the negligence in pedestrian's behavior or unawareness during the crossing of the road [3]. Effect of falls can be broadly classified as: (i) physical harm (e.g., fractures and cuts), (ii) psychological repercussions (e.g., loss of independence), (iii) financial burden, and (iv) death (mostly applicable to elder person). It is essential to develop a model that can be used in detecting real-time falls on road to mitigate their impacts by sending the immediate help to the fall locations, thereby enhancing the pedestrian safety.

Various sensory modalities, such as accelerometer [4], inertial measuring units (IMUs) [5], microphone [6], radar [7], and camera [8] are used for gathering the information of falls. Accelerometers can capture the information of change in acceleration and velocity. Whereas, IMUs provide more information on movement and orientation by combining accelerometers, gyroscopes, and occasionally magnetometers. Microphones can be used to capture sounds which are occurred during falls, including screams or the sound of an impact. Radar sensors and cameras are used for capturing falls in terms of electromagnetic waves and images, respectively. Out of these, cameras are cheaper smaller and of higher quality than ever before. Moreover, imaging technology

* Corresponding author.
E-mail addresses: apramanik17@gmail.com (A. Pramanik), soumicksarker9@gmail.com (S. Sarker), sobhan.sarkar@iimranchi.ac.in, sobhan.sarkar@gmail.com (S. Sarkar), sankarpal@yahoo.com (S.K. Pal).

https://doi.org/10.1016/j.knosys.2025.113038
Received 24 June 2024; Received in revised form 5 November 2024; Accepted 16 January 2025
Available online 2 February 2025

has immensely progressed in recent years. Concurrently, the computing power of multi-core processor has dramatically increased. All these lead to using cameras for fall detection to pursue real-time applications [9].

Falls can be occurred in both outdoor and indoor areas. Outdoor areas include construction site and road [10]. Whereas, indoor areas include home and hospital [10]. Image-based analysis concentrates on specific details of human posture, movement, and orientation to increase the fall detection accuracy. Image analysis can be done using both machine vision and image processing. Machine vision-based models can perform complex tasks. Nowadays, the existing large datasets (containing various situations) make machine vision-based models highly flexible and adaptable. Moreover, machine vision-based models can integrate multiple data types, such as visual data, sensor data, and contextual information. Machine vision-based models can provide a more comprehensive understanding of complex situations, thereby improving the accuracy. All these prove the superiority of machine vision-based models over image processing-based models for fall detection [11]. Machine vision-based models for fall detection are broadly classified into two categories, including machine learning-based models and deep learning-based models [10]. In recent times, due to having the self-learning ability using suitable features, deep learning-based models (computer vision (CV)) provide their superiority over traditional machine learning-based models [11]. Patterns of subject's posture and movement are very useful for characterizing fall scenarios [12,13]. In case of unavailability of data, Generative Adversarial Network (GAN) [14] is used for generating the fall data which is further used for training, thereby enhancing the precision. Convolutional Pose Machine (CPM) [15] is one of the most often utilized multi-stage deep networks for fall detection. CPM is used to determine the human body's 2D stance from an image or video frame. But it requires high processing power. Deep Convolutional Neural Network (DCNN)-based object-level features are temporally aggregated using Long Short-Term Memory (LSTM) network for fall detection [16]. In [17], the gated recurrent unit (GRU) is combined with CNN for fall detection. Context-aware model-based feature(s) analysis using 3D-CNN provides higher accuracy in fall detection tasks [18].

Aforesaid deep learning-based models are restricted to the huge amount of fall incident related data, which may not be available or affordable always. Moreover, aforesaid studies are focused on designing models or networks for detecting either indoor falls (related to senior citizens residing at home) or outdoor falls (related to workers who are working in the construction area). This is due to the nature of falls which is varied from one scenario to another. It creates uncertainty in describing the characteristics of falls. The said uncertainty issue should be addressed to detect falls in both indoor and outdoor scenarios and differentiate the fall event from the no-fall event. All these necessitate to make a generic fall detection system that can be applicable to detect both indoor and outdoor falls concerning complex scenarios. Granular computing [19] which function in granular level for handling uncertainty is effective for large data. Various techniques such as fractional fuzzy grammar [20], fuzzy geometry [21] and rough-fuzzy integration [22] are developed in handling such uncertainty arising from granularity. While the fractional fuzzy grammar deals with spatial geometrical granularity in image plane for its syntactic recognition, fuzzy geometry combines the fuzzy set theory with approximate reasoning in handling the imprecise patterns of data. The rough-fuzzy integration [22] is significant when the uncertainty arises from both overlapping region and granularity in the domain. Due to the advantages of granulation in uncertainty analysis, we have used the concept of granulation in defining the approximate regions of falls.

It is well known that LSTM-based models are designed to capture the deviation in temporal information for a long sequence [16,23]. It is also known that 'Fall' is not an instantaneous event. A 'Fall' event involves human movements over time. Due to having the ability to model the temporal progression for long sequences, LSTM-based models are superior to traditional object detectors in distinguishing falls from no-fall activities. Characteristics of the 'Fall' event are varied from one scenario to another. Therefore, it is hard to obtain labeled data containing fall scenarios. In cases of data scarcity problems, transfer learning is used to retrain the model efficiently [24]. Using transfer learning, the model's performance can be improved significantly in cases of limited training data. By incorporating the concept of transfer learning within the LSTM, fall detection accuracy can be improved for the data scarcity problem. The traditional LSTM-based fall detection models use the entire region as a region of interest (RoI) for classification. In case of a data scarcity problem, some undesirable regions (e.g., 'No Fall' regions) might affect the detection accuracy for their wrong classification. To overcome this issue, the concept of granulation (clustering) is introduced within the LSTM to narrow its focus on the regions where the probability of occurrence of falls is higher than that in any other parts, thereby increasing both speed and accuracy. Keeping in mind the advantages of using granulation, LSTM, and transfer learning for detection tasks in the temporal domain with data scarcity problems, a new model, namely transfer learning-based granulated Bi-LSTM (TLG-LSTM) is developed using the aforesaid concepts for fall detection.

The developed TLG-LSTM consists of four stages: (i) object detection and tracking, (ii) feature extraction from the detected objects, (iii) feature analysis for granulation, and (iv) transfer learning-based Bi-LSTM for granule classification as 'Fall' or 'No Fall'. For object detection, we employ a combination of YOLOv8 (one-stage detector) [25] and Faster RCNN (two-stage detector) [26] to maintain a trade off between inference speed and detection accuracy. After detection, bounding boxes are fitted over all the detected objects (persons) to track individuals. Subsequently, for each tracked object, both MoveNet-Lightning and object-level features are computed and investigated for better modeling of falls in diverse scenarios. MoveNet-Lightning features are leveraged for human pose estimation, trained on the MSCOCO Dataset. Two MoveNet-Lightning features, namely head-hip distance and hip-ankle distance, and three object-level features, namely change in aspect ratio, speed variation, and change in area are used. As GPU may not be afforded always, the CPU version of MoveNet-Lightning-based pose estimator is used to make the feature extraction process more energy efficient and cost effective. Both MoveNet-Lightning and object-level features are analyzed for accessing granules that represent the approximate regions concerning falls. These granules are fed to the Bi-LSTM network to obtain the specific fall location. Granulation using both MoveNet-Lightning and object-level features, along with Bi-LSTM classification tasks, strengthens the detection results. It is well known that deep network requires enormous training data which may not be accessible always. In order to address this issue, we have used temporal self attention mechanism-based transfer learning for re-training of Bi-LSTM even in case of unavailability of data.

The proposed TLG-LSTM is demonstrated over several real-time videos comprising of various 'Fall' and 'No Fall' events concerning both outdoor (road) and indoor (home) scenarios. These videos are acquired from URFD and YouTube8M datasets [10]. To show the superiority of the developed TLG-LSTM, a comparative study is done with some other state-of-the-art models. It is also demonstrated that the developed TLG-LSTM is able to overcome other issues, including generalization ability in applying to multiple situations with real-life complex scenarios. Thus, the objective of this study is to develop a robust multi-person fall detection model that can be used in a variety of complex scenarios. Based on the aforesaid discussion, the contributions of our study can be summarized as below:

(i) A new model, namely transfer learning-based granulated Bi-LSTM (TLG-LSTM) is developed for real-time fall detection in various scenarios concerning both indoor (i.e., home) and outdoor (i.e., road) areas.
(ii) Unlike state-of-the-art models, in TLG-LSTM, both MoveNet-Lightening and object-level features are analyzed for characterizing the traffic-fall scenarios in an improved way.

(iii) In TLG-LSTM, three object-level features, namely change in aspect ratio, speed variation, and change in area, and two MoveNet-Lightening pose features, namely head-hip distance and hip-ankle distance are analyzed for obtaining granules by defining two linguistic rules. Commonality between these two granules represents the approximate locations of falls.
(iv) Common granules are fed to the Bi-LSTM network for detecting the specific locations of falls. Instead of the entire frame, use of granules for fall classification, enhances the detection speed as well as accuracy.
(v) For re-training of Bi-LSTM, temporal self attention mechanism-based transfer learning is used to enhance the training accuracy and speed.

The rest of the paper is organized as follows: Section 2 presents the related works. The developed TLG-LSTM is illustrated in Section 3. In Section 4, the results are discussed. Finally, we conclude this study and state some future scopes in Section 5.

2. Related works

The key literature on fall detection are broadly categorized into two classes, including rule-based models and deep learning-based models, as described in the following sections.

2.1. Rule-based models

In rule-based models, useful features are computed and evaluated using some defined rules (based on thresholds) for a specific task. This is done in an unsupervised way, therefore, rule-based models are useful in case of having less amount of training data [10]. The effectiveness of a rule-based models depends on how the features and corresponding thresholds are chosen [27]. Features used for fall detection are broadly categorized into two classes, namely pose features and object-level features. Pose estimation models are used to obtain the pose of human based on the pixel-related features [28]. In [29], human skeleton related OpenPose features are extracted and used for rule generation for classifying an event as 'No Fall' or 'Fall'. In [30], the MoveNet-Lightening features are used to train the MobileNetV2 network for detecting falls in aircraft maintenance environment. Lightweight pose estimation technique is used in [31] for real-time fall detection. In [10], it is demonstrated that the pose features combined with object-level features provides better modeling of the fall events even in case of complex scenarios.

A large number of researches have concentrated on object-level features to model the characteristics of human falls. In [32], center of gravity and aspect ratio of each detected person are used for fall detection. In another study [33], height of centroid and torso angle are used for fall detection. The silhouette area ratio is also effective in fall detection [34]. For a person, volume is combined with the waiting time spent by its centroid for modeling fall event [35]. In [36], various object-level features, context, and semantic information are fed to the deep network having multiple branches for characterizing human falls. In [10], Z-numbers computation along with rule-based feature analysis are done for modeling fall events concerning both outdoor and indoor scenarios. The formation of rough fuzzy granules using these features may enhances the detection accuracy. Some key literature on deep learning-based fall detection models are presented in the next section.

2.2. Deep learning-based models

In recent years, deep learning-based models gain a lot of attention due to their self learning ability [3]. In [37], deep CNN is integrated with LSTM for fall detection by aggregating the temporal features. In [38], the deep CNN is combined with GRUs for fall detection by pooling latent features. In [39], the LSTM is trained using OpenPose features for detecting human falls. Whereas, in [40], 3D-CNN is trained with depth information for recognizing an event as 'Fall' and 'No Fall', leading to inaccurate results. In [41], Recurrent Neural Network (RNN) is integrated with LSTM for fall detection. Due to having the ability of data coding for unsupervised models, auto-encoders gain much attention in fall detection tasks [42]. The spatio-temporal feature-based convolutional auto-encoder is developed in [43] for detecting human falls. In [17], OpenPose and skeleton features are used to train the LSTM network for modeling human falls.

Using transfer learning, the pre-trained model which is trained over general datasets, can be applied to a specific task. This significantly reduces the amount of domain-specific data requirement. Therefore, transfer learning is very effective for data scarcity problems [44]. Several studies have explored the use of transfer learning for fall detection tasks. In [4], transfer learning is used to retrain a deep CNN (having three convolutions, two max pooling, and three fully connected layers) for fall detection over IoT and fog computing environments. In [18], transfer learning is used to a pre-trained 3D CNN model (trained over MS-COCO dataset) for capturing the spatio-temporal patterns of fall events. In [45], the knowledge of human body movements is used to fine-tune the pre-trained deep CNN for fall detection. In [46], transfer learning is done by adding batch normalization and drop-out layers to a simple deep CNN (called HActivityNet) for human fall detection. In [47], a new Two-Stream Inflated 3D ConvNet (I3D) is developed based on 2D ConvNet inflation, i.e., filters and pooling kernels of very deep image classification ConvNets are expanded into 3D. The I3D architecture is developed for general action recognition. It is adopted for fall detection tasks by re-training on Kinetics data. In [48], a multi-stage architecture, namely MS-TCN is developed for fall detection due to its ability to multi-scale action in video sequences. However, aforesaid studies are focused on either indoor falls (occlusion-free scenarios) constructed for senior citizens living in homes or outdoor falls (combination of both occlusion-free and complex scenarios) developed for workers working in the construction area. Moreover, the uncertainty issue that arises among various 'Fall' and 'No Fall' events corresponding to the complex scenarios may not be handled using the aforesaid deep learning-based studies.

Based on the aforesaid discussion, key research issues are mentioned as follows:

(i) There is a very limited research on using granulation for fall detection tasks.
(ii) Deep learning-based models are focused on either indoor falls or outdoor falls.
(iii) There is a lack of research on deep learning-based models for handling the uncertainty issue arising between 'Fall' and 'No Fall' events.

To address the aforesaid issues, transfer learning-based granulated Bi-LSTM (TLG-LSTM) is developed by incorporating the concept of granulation, pose-features, transfer learning, and Bi-LSTM, as described in the next section.

3. TLG-LSTM

The developed TLG-LSTM model is demonstrated in Fig. 1. From this figure, it is seen that, initially, objects (i.e., person) are detected and tracked for each input frame, as described in Section 3.1. Thereafter, both MoveNet-Lightening and object-level features are computed corresponding to each detected and tracked object, as explained in Section 3.2. Aforesaid features are analyzed for obtaining granules, as presented in Section 3.3. Instead of the entire frame, only these granules are fed to the Bi-LSTM network for classification tasks. Temporal self attention mechanism-based transfer learning is used to re-train the Bi-LSTM network, as explained in Section 3.4.

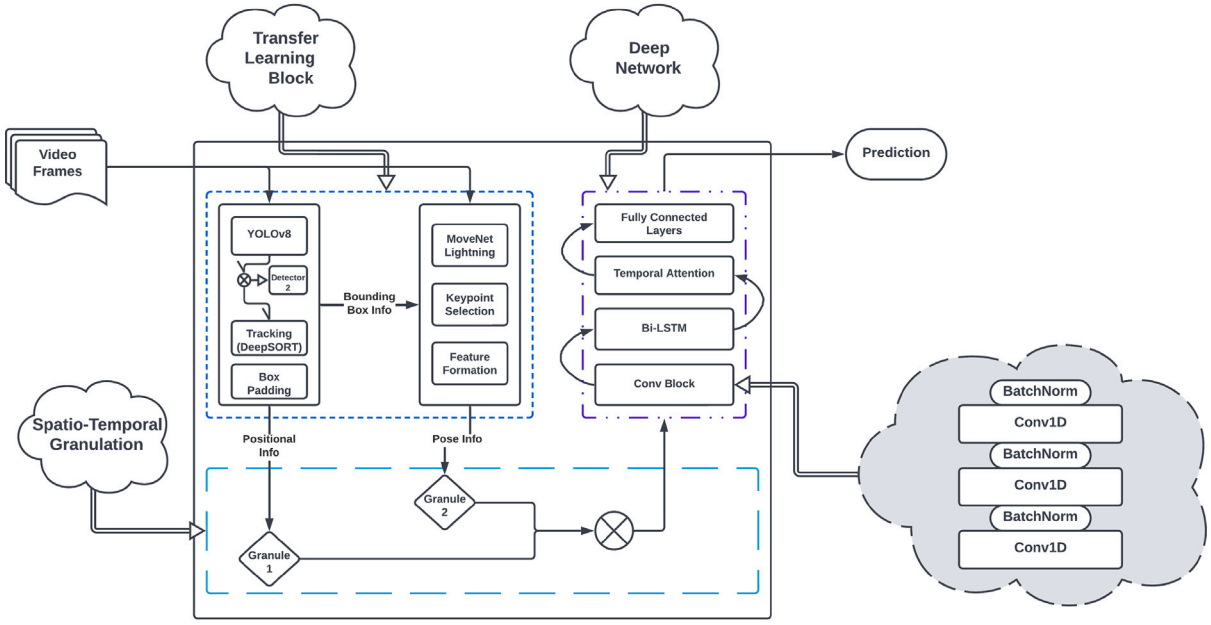


Fig. 1. A schematic diagram of TLG-LSTM.

3.1. Object detection and tracking

Object tracking is followed by the object detection task. Methods used for object detection and tracking are stated in Sections 3.1.1 and 3.1.2, respectively.

3.1.1. Object detection

Fall event is related to only one object-class, i.e., person. Therefore, person detection is the pre-requisite of fall detection tasks. It is well known that one-stage detectors can achieve high inference speed [25] and two-stage detectors can achieve high accuracy [11] in object detection tasks. It is also seen from an experiment that, one-stage detectors while performing admirably under typical conditions, experience interruptions in detection when individuals assume unconventional postures during a fall event. The detection using one-stage detector resumes once the individual is lifted and regained a standing position, resulting an information loss during these intervals. Due to having the region proposal block, two-stage detectors can address the aforesaid issue but slows down the process. Keeping in mind the advantages of both detectors, we have combined both one-stage and two-stage detectors for obtaining a trade off between speed and accuracy in handling such complex situations to ensure comprehensive fall detection coverage. Due to having high inference speed, one-stage detector is used as primary detector for fitting the bounding-boxes over the detected objects present in a frame. Whereas, two-stage detector is used as a secondary detector which is used for two cases: (i) when the bounding-box obtained by the primary detector is disappeared in an intermediate frame, and (ii) mis-classification is done by the primary detector.

Various one-stage detectors, namely YOLOv5, YOLOv7, and YOLOv8 [25] have demonstrated commendable performance in terms of both inference speed(ms) and accuracy (mAP). A comparative study is conducted to find the best one among these three detectors. All these detectors are trained on the MS COCO dataset and tested over 500 frames that are acquired from a URFD video [49]. Detection results are shown in Table 1. From this table, it is evident that YOLOv8 is superior in terms of speed, mAP, and % of false detection. Hence, the detector, YOLOv8 is used as a primary detector. Whereas, a two-stage-detector, namely Faster R-CNN [26] is used as a secondary detector due to its high detection accuracy. ResNet-50 is used as a backbone of Faster R-CNN. The combination of YOLOv8 and Faster RCNN exhibits improved performance when confronted with awkward poses. The result of YOLOv8+Faster RCNN is presented in Fig. 2. Here, Fig. 2(a) represents the detection result of YOLOv8. From this figure, it is seen that YOLOv8 successfully detects person along with some mis-classification of background as person. In order to eliminate these mis-classified background, corresponding regions are fed to the Faster RCNN for obtaining the final detection result, as presented in Fig. 2(b). This result evident that all mis-classifications are eliminated. Therefore, in this study, we have used the combination of YOLOv8 and Faster RCNN for person detection. After the detection, detected persons are tracked, as explained in the next section.

Table 1
YOLO Comparisons on 500 frames (GPU: RTX 3060 6GB).

	YOLOv5	YOLOv7	YOLOv8
Speed (ms)	2150	1072	913
False Detection	22.9%	7.3%	8.2%
mAP	0.42	0.47	0.48

3.1.2. Object tracking

Among various tracking techniques, Deep SORT demonstrates the fastest performance, achieving an average speed of 24 fps (frames per second) while maintaining high accuracy [11]. Therefore, Deep SORT is used to facilitate continuous tracking of individuals across a sequence of frames. Deep SORT employs a convolutional deep network to generate an appearance descriptor corresponding to the detected object. This appearance descriptor is subsequently utilized for tracking through Kalman Filters. During the detection task, when an object initially detected in a frame, it is assigned by a unique identification (ID). During tracking, this ID is then employed for making an assignment between the same object appears in subsequent frames. It is also found from an extensive study that stationary objects which are not indicative to a fall, may not introduce variations in feature shapes across different batches of frames. During tracking, for obtaining these stationary objects, two approaches are implemented: (i) if an object appears for only one frame corresponding to a batch, it is considered as stationary object from the frame where it is detected first time, and (ii) any object exiting the frame before the final frame of the batch is considered stationary from the frame of its last appearance until the last frame of the batch. For each detected and tracked object (person), various features are computed, as explained in the following section.

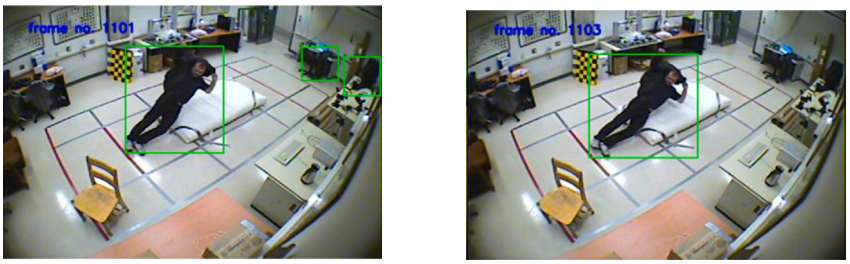


Fig. 2. Results of YOLOv8+Faster R-CNN.

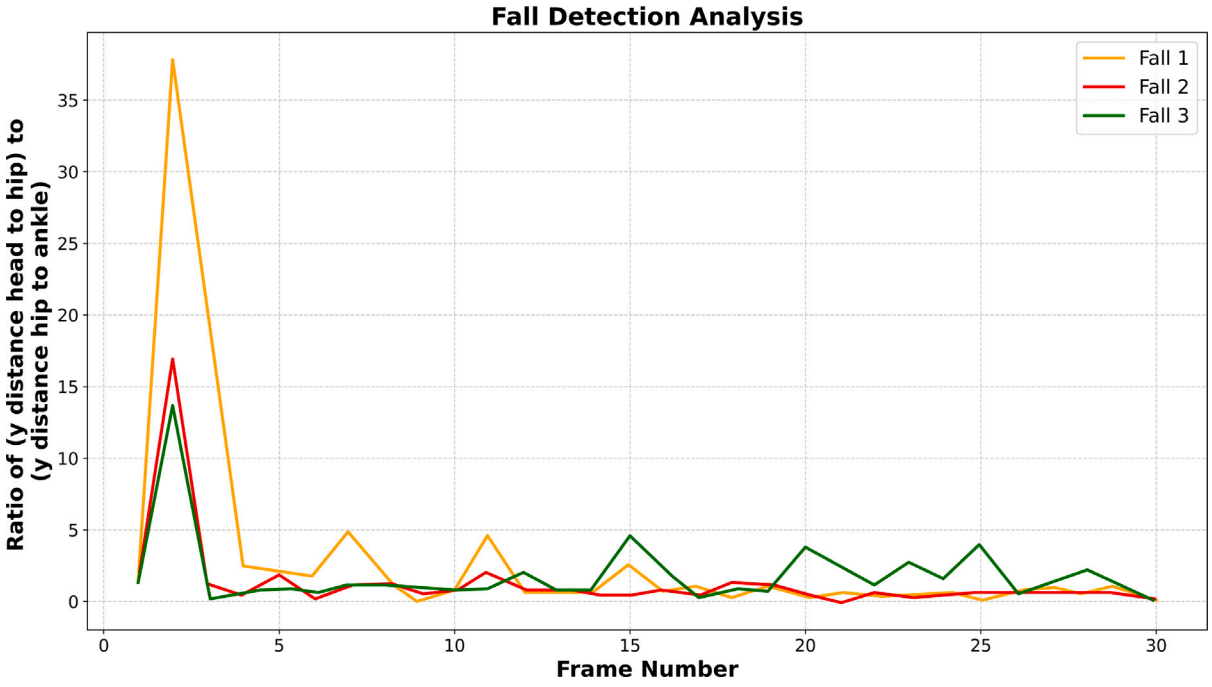


Fig. 3. Co-relation between position of head, hip, and ankle.

3.2. Feature computation

Both pose and object-level features are computed corresponding to each detected and tracked object (person). Various well-known pose estimators are OpenPose, MoveNet-Lightening, MoveNet-Thunder, and PoseNet. Total time required by all these pose-estimators for processing 500 frames are 283.453 s, 28.446 s, 64.933 s, and 39.831 s, respectively. It is evident that, OpenPose performs the slowest while MoveNet-Lightning performs the fastest. While testing on MS-COCO dataset it is found that there is a comparable accuracy between PoseNet and MoveNet-Lightening. To maintain a trade off between speed and accuracy, MoveNet-Lightening is used in this study for estimating the position of head, hip, and ankle of a person instead of all the joints. For three fall scenarios, ratios between head-hip distances and hip-ankle distances over thirty consecutive frames are presented in Fig. 3. From this figure, it is evident that, a definite trend (high spike) of ratios (between head-hip and hip-ankle distances) is found from 1st to 5th frames corresponding to an actual fall event. Therefore, three pose features, namely position of head, position of hip, and position of ankle are used in modeling fall events.

Let o_i^t be the ith detected object (person) present in the current frame (f_t). Let $y_{head_i^t}$, $y_{hip_i^t}$, and $y_{ankle_i^t}$ be the locations of y-coordinate corresponding to 'head', 'hip', and 'ankle' for o_i^t. Let $x_{head_i^t}$, $x_{hip_i^t}$, and $x_{ankle_i^t}$ be the locations of x-coordinate corresponding to 'head', 'hip', and 'ankle' for o_i^t. Let $d(HH_i^t)$ and $d(HA_i^t)$ be the head-hip and hip-ankle distances for o_i^t, defined as:

$$\begin{aligned} d(HH_i^t) &= \frac{\left|y_{head_i^t} - y_{hip_i^t}\right| + \left|x_{head_i^t} - x_{hip_i^t}\right|}{2} \\ d(HA_i^t) &= \frac{\left|y_{hip_i^t} - y_{ankle_i^t}\right| + \left|x_{hip_i^t} - x_{ankle_i^t}\right|}{2} \end{aligned} \tag{1}$$

For each detected person, along with pose features, three object-level features, namely speed variation, change in area, and change in aspect ratio are considered. Aspect ratio is the ratio between width and height of the bounding-box fitted over the detected person. Let AR_i^t, V_i^t, and A_i^t be the aspect ratio, velocity, and area for the ith person (o_i^t) present in f_t. Let B_i^t be the bounding-box fitted over o_i^t. Let (x_i^t, y_i^t) be the location of the centroid corresponding to B_i^t. Let w_i^t and h_i^t be the width and height of B_i^t. Here, AR_i^t, V_i^t, and A_i^t are defined as:

$$\begin{aligned} AR_i^t &= \frac{w_i^t}{h_i^t} \\ V_i^t &= \sqrt{(Vx_i^t + Vy_i^t)} \\ A_i^t &= w_i^t \times h_i^t \end{aligned} \tag{2}$$

where Vx_i^t and Vy_i^t represent velocity components for o_i^t corresponding to the x-axis and y-axis, respectively. Velocity component is measured based on the spatial information of consecutive three frames, as defined: $Vx_i^t = \frac{\left|x_i^{t-3} - x_i^t\right|}{3}$ and $Vy_i^t = \frac{\left|y_i^{t-3} - y_i^t\right|}{3}$.

All these object-level features and pose features are analyzed for obtaining granules that are prone to fall locations, as explained in the next section.

3.3. Granulation

Granulation is done to obtain the probable fall regions over the video frames in terms of granules (clusters). Two types of granules, namely object-granule and pose-granule are formed using object-level features and pose features, respectively. The object-granule ($\{OG\}$) is formed using the information of aspect-ratio, velocity, and area. There are two types of object granules, namely $\{OG_f\}$ and $\{OG_{nf}\}$ corresponding to 'Fall' and 'No Fall' scenarios, respectively. During the fall incident, there is an abrupt changes found in aspect ratio, velocity, and area. Let $d(AR_i^t)$, $d(V_i^t)$, and $d(A_i^t)$ be the change in aspect ratio, velocity, and area, respectively. The $d(AR_i^t)$, $d(V_i^t)$, and $d(A_i^t)$ are defined as:

$$\begin{aligned} d(AR_i^t) &= \left|AR_i^t - AR_i^{t-1}\right| \\ d(V_i^t) &= \left|V_i^t - V_i^{t-1}\right| \\ d(A_i^t) &= \left|A_i^t - A_i^{t-1}\right| \end{aligned} \tag{3}$$

An object, o_i^t belongs to $\{OG_f\}$, if it follows the rule which is defined below.

$$o_i^t \in \{OG_f\} \mid (d(AR_i^t) > Th_f^1) \cap (d(V_i^t) > Th_f^2) \cap (d(A_i^t) > Th_f^3) \neq 0 \tag{4}$$

where thresholds, Th_f^1, Th_f^2, and Th_f^3 are defined as:

$$\begin{aligned} Th_f^1 &= \mu_{AR} + \sigma_{AR} \\ Th_f^2 &= \mu_V + \frac{1}{2}\sigma_V \\ Th_f^3 &= \mu_A + \frac{1}{2}\sigma_A \end{aligned} \tag{5}$$

where $\mu_{()}$ and $\sigma_{()}$ define mean and standard deviation of a set, say $()$ containing a feature (either object feature or pose feature) for an object over all frames (P) present in a video containing both 'Fall' and 'No Fall' scenarios. An object, o_i^t belongs to $\{OG_{nf}\}$, if it follows the rule which is defined below.

$$o_i^t \in \{OG_{nf}\} \mid (d(AR_i^t) < Th_{nf}^1) \cap (d(V_i^t) < Th_{nf}^2) \cap (d(A_i^t) < Th_{nf}^3) \neq 0 \tag{6}$$

where thresholds, Th_{nf}^1, Th_{nf}^2, and Th_{nf}^3 are defined as:

$$\begin{aligned} Th_{nf}^1 &= \mu_{AR} + \frac{1}{2}\sigma_{AR} \\ Th_{nf}^2 &= \mu_V + \frac{1}{2}\sigma_V \\ Th_{nf}^3 &= \mu_A + \frac{1}{2}\sigma_A \end{aligned} \tag{7}$$

The defined granules, $\{OG_f\}$ and $\{OG_{nf}\}$ represent the probable locations of falls and no-falls, respectively. Another granule, namely pose-granule ($\{PG\}$) is formed using the information of head-hip and hip-ankle distances. There are two types of pose granules, namely $\{PG_f\}$ and $\{PG_{nf}\}$ corresponding to 'Fall' and 'No Fall' scenarios, respectively. During fall, there is an abrupt change found in head-hip distance and hip-ankle distance. An object, o_i^t belongs to $\{PG_f\}$, if it follows the rule defined below.

$$o_i^t \in \{PG_f\} \mid \frac{d(HH_i^t)}{d(HA_i^t)} > Th_f^4 \tag{8}$$

where thresholds, Th_f^4 is defined as: $Th_f^4 = \mu_{\frac{d(HH)}{d(HA)}} + \sigma_{\frac{d(HH)}{d(HA)}}$. An object, o_i^t belongs to $\{PG_{nf}\}$, if it follows the rule which is defined below.

$$o_i^t \in \{PG_{nf}\} \mid \frac{d(HH_i^t)}{d(HA_i^t)} < Th_{nf}^4 \tag{9}$$

where thresholds, Th_{nf}^4 is defined as: $Th_{nf}^4 = \mu_{\frac{d(HH)}{d(HA)}} - \sigma_{\frac{d(HH)}{d(HA)}}$. The defined pose-granules, $\{PG_f\}$ and $\{PG_{nf}\}$ represent the probable location of falls and no-falls, respectively. In order to get the higher degree of reliability for the probable fall location, a new granule, namely $\{CG\}$ is obtained by keeping the commonality between $\{OG\}$ and $\{PG\}$. There are two types of $\{CG\}$, namely $\{CG_f\}$ and $\{CG_{nf}\}$ for representing the approximate Fall and No Fall regions. The $\{CG_f\}$ is defined as:

$$o_i^t \rightarrow \{CG_f\} \mid \frac{\{OG_f\} \cap \{PG_f\}}{\{OG_f\} \cup \{PG_f\}} > 0.5 \tag{10}$$

Whereas, the $\{CG_{nf}\}$ is defined as:

$$o_i^t \rightarrow \{CG_{nf}\} \mid \frac{\{OG_{nf}\} \cap \{PG_{nf}\}}{\{OG_{nf}\} \cup \{PG_{nf}\}} > 0.5 \tag{11}$$

Instead of the entire frame, the developed $\{CG_f\}$ and $\{CG_{nf}\}$ are used for classification tasks. This is explained in the next section.

3.4. Classification network

The developed $\{CG_f\}$ and $\{CG_{nf}\}$ are fed to the classification network for fall detection. As seen in Fig. 1, the classification network consists of three parts: Conv block, Bi-LSTM, and Fully connected layers. The Conv block has three linear Conv1D layers, namely $Conv1D_1$, $Conv1D_2$, and $Conv1D_3$ which contain 32 filters, 128 filters, and 256 filters, respectively. Each filter having size of (3×3). The output of $Conv1D_3$ layer is fed to the Bi-LSTM network [50] of having 512 hidden nodes. The output of Bi-LSTM is converted to 1-dimensional weighted array through the fully connected layer. Thereafter, this 1-dimensional weighted array is fed to the SVM for classification tasks. Two-class ('Fall' and 'No Fall') prediction is done. In case of having insufficient training data, to re-train the Bi-LSTM network, the concept of transfer learning is introduced with the Bi-LSTM network.

Transfer learning using the temporal self attention mechanism facilitates the understanding and capture of the underlying patterns required for accurate prediction by navigating through the sequence data with judicious allocation of focus to key time steps. This mechanism can be easily integrated into many deep learning architectures to improve their ability to capture temporal dependencies. Therefore, temporal self attention mechanism-based transfer learning is used to re-train the Bi-LSTM network. The temporal self attention mechanism directs the network's attention to the key parts of the input sequence that are most pertinent at each time step, while dynamically altering the importance of various input pieces in accordance with their relevance to the current output. Each time step in the input sequence receives a weight from the temporal self attention mechanism based on how important it is to the present output. Let H and $X = (\{CG_f\}, \{CG_{nf}\})$ be the current hidden state and the complete input sequence to the Bi-LSTM network. The softmax function is applied over H and X to compute a set of weights, say $\{Aw\}$, defined as:

$$\{Aw\} = softmax\left(\{X\}^T \cdot \{H\}\right) \tag{12}$$

where T represents the transpose of a matrix. At each time step, a context vector, say $\{Cw\}$ is created based on the weighted sum of the input sequence elements and the set of weights, $\{Aw\}$. The $\{Cw\}$ is defined as:

$$\{Cw\} = \{H\} \cdot \{Aw\} \tag{13}$$

The most important elements of the input sequence at each time step are captured by the attention vector, which is subsequently used to determine the output. The pre-activation vector, say $\{Pv\}$ is defined as:

$$\{Pv\} = \{Cw\} + \{H\} \tag{14}$$

Using the pre-activation vector, attention vector ($\{Av\}$) is defined as:

$$\{Av\} = softmax(\{Pv\}) \tag{15}$$

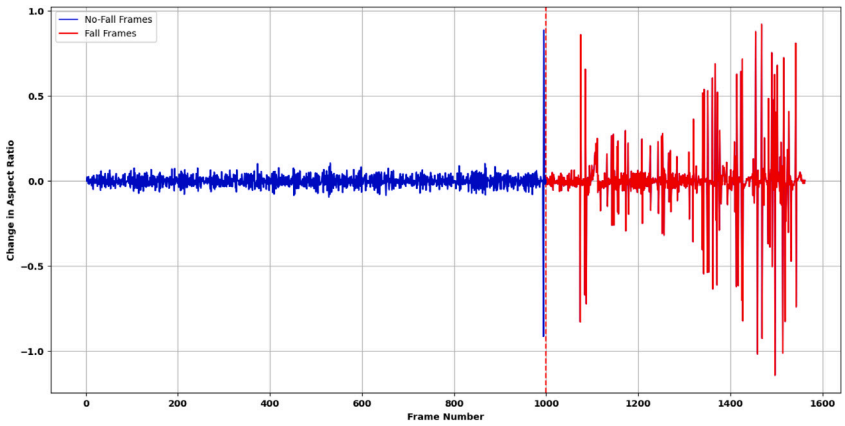


Fig. 4. Change in aspect ratio over frames.

The output of Bi-LSTM is computed based on $\{Av\}$, defined as:

$$output = tanh(\{Av\}) \tag{16}$$

The temporal self attention mechanism is used in this study to manage the lengthy input sequence by concentrating only on its (sequence's) most important elements, thereby increasing the speed as well as accuracy. Using the temporal self attention mechanism-based transfer learning, the influence of different input sequence segments over the network's predictions is interpreted in better way. Experimental results are discussed in the following section.

4. Result & discussion

Our study has three broad sub-objectives: (i) to demonstrate the effectiveness of granulation over Bi-LSTM for real-time fall detection, (ii) to demonstrate the utility of transfer learning for lengthy input sequence, (iii) to demonstrate the superiority of TLG-LSTM over some state-of-the-art models. The details of used datasets, experimental set up, and experimental results are discussed in Sections Section 4.1, 4.2, and 4.3, respectively.

4.1. Data description

Exiting fall datasets contain either simulated data or real-life fall scenarios. Here, experts design fall scenarios for generating the simulated data. Whereas, in case of real-life scenario, the nature of fall may vary in two consecutive frames. Therefore, simulated falls may be different than real-life falls. Thus, the knowledge of simulated falls may not be used for training and(or) validating a model. The effectiveness of the developed TLG-LSTM is assessed over two real-life video datasets, namely URFD [49] and YouTube8M (YT8M) [10]. Each training and(or) test video contains diverse visual entities. From URFD, twelve labeled videos concerning indoor scenarios (containing 'Fall' and 'No Fall' events) are used for training and testing of the developed model. Whereas, YouTube8M contains various large-scaled unlabeled traffic videos. From YouTube8M, a total of fourteen real-life traffic videos containing both 'Fall' and 'No Fall' events are considered. Annotation is not done with YouTube8M videos, therefore, YouTube8M data is used for testing process only to hold the universality of the TLG-LSTM. The test videos that are acquired from YouTube8M contain 350 to 1245 frames with a variety of real-life traffic scenarios, including occlusion, multi-falls, and no fall. A detailed description of experimental setup is presented in the next section.

4.2. Experimental setup

Algorithms are coded in Python 3.6 (Anaconda), with a multi-core AMD Ryzen 7 CPU and an NVIDIA RTX 3060 GPU. Three libraries, namely cv2, TensorFlow, and Numpy are used. As defined in Section 3, the developed TLG-LSTM contains four parts, namely object detection and tracking, feature computation, granulation, and temporal self attention mechanism-based Bi-LSTM for classification tasks. The detector, namely YOLOv8+Faster RCNN and tracker, namely Deep SORT are used for object (person) detection and tracking, respectively. The YOLOv8 is trained on MSCOCO data [51]. Standard parameters of YOLOv8 [52] and DeepSORT [53] are used in this study. It is also found in Section 3.2, both MoveNet-Lightening and object-level features are extracted from the detected person. These features are analyzed for obtaining the approximate regions of falls in terms of granules (refer to Section 3.3). The parametric study for obtaining optimum granules is presented in Section 4.2.1. As already discussed in Section 3.4, these optimum granules are fed to the granulated Bi-LSTM for obtaining the exact fall location. As already mentioned, transfer learning is done using the temporal self attention mechanism. The parametric study for obtaining the optimum parameters of transfer learning-based granulated Bi-LSTM is presented in Section 4.2.2.

4.2.1. Parametric study for obtaining optimum granules

As already mentioned, two types of granules, namely object granule ($\{OG\}$) and pose granule ($\{PG\}$) are defined using object-level features and MoveNet-Lightening features, respectively. Here, $\{OG_f\}$ is developed using the information of change in aspect ratio, speed variation, and change in area based on three thresholds, namely Th_f^1, Th_f^2, and Th_f^3, respectively. Similarly, $\{OG_{nf}\}$ is developed based on three thresholds, namely Th_{nf}^1, Th_{nf}^2, and Th_{nf}^3. A total of ten videos containing both indoor and outdoor scenarios are used for obtaining the optimum values of aforesaid thresholds. The mean distributions of change in aspect ratio for 'Fall' and 'No Fall' events corresponding to a video of outdoor scenario are shown in Fig. 4. From this figure, it is seen that $\mu^{AR} = 0.09$, $\sigma_{nf}^{AR} = 0.083$, $\sigma_f^{AR} = 0.407$. For, considering ten annotated videos, average of ten μ^{AR} and σ^{AR} are 0.1 and 0.21, respectively. Therefore, for change in aspect ratio, two thresholds, $Th_f^1 = 0.31$ and $Th_{nf}^1 = 0.115$ are selected. The mean distributions of speed variation for 'Fall' and 'No Fall' events corresponding to the same video are shown in Fig. 5. From this figure, it is seen that for 'No Fall' event, $\mu_{nf}^V = 3.265$ and $\sigma_{nf}^V = 14.17$. It is also found that, for 'Fall' event, $\mu_f^V = 46.82$ and $\sigma_f^V = 133.1$. For the entire video having both 'Fall' and 'No Fall' scenarios, $\mu^V = 25.82$ and $\sigma^V = 73.63$. Considering ten annotated videos, average of ten μ^V and σ^V are 27.4 and 74.16, respectively. Therefore, for speed variation, two thresholds, $Th_f^2 = 64.48$ and $Th_{nf}^2 = 64.48$ are selected. The mean distributions

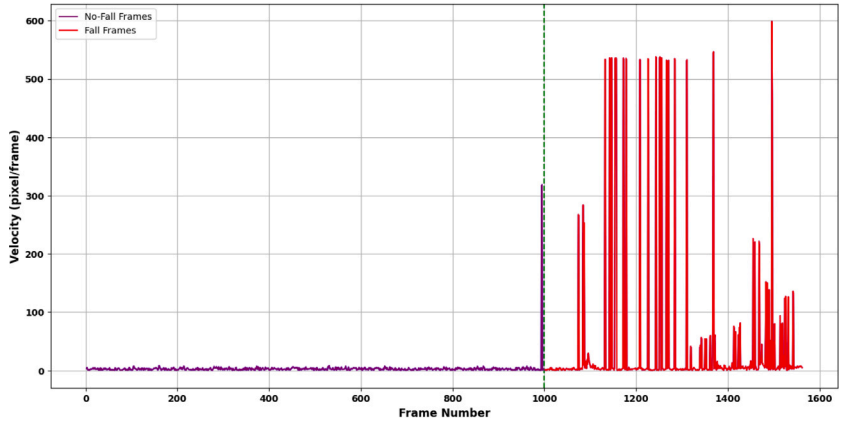


Fig. 5. Speed variation over frames.

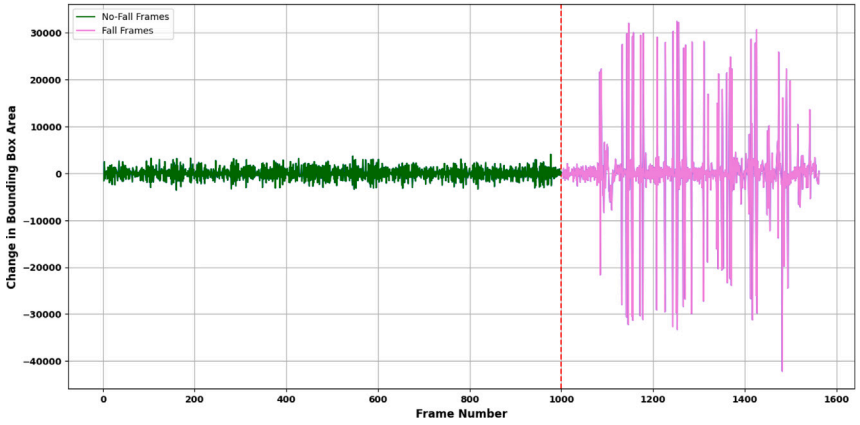


Fig. 6. Change in area over frames.

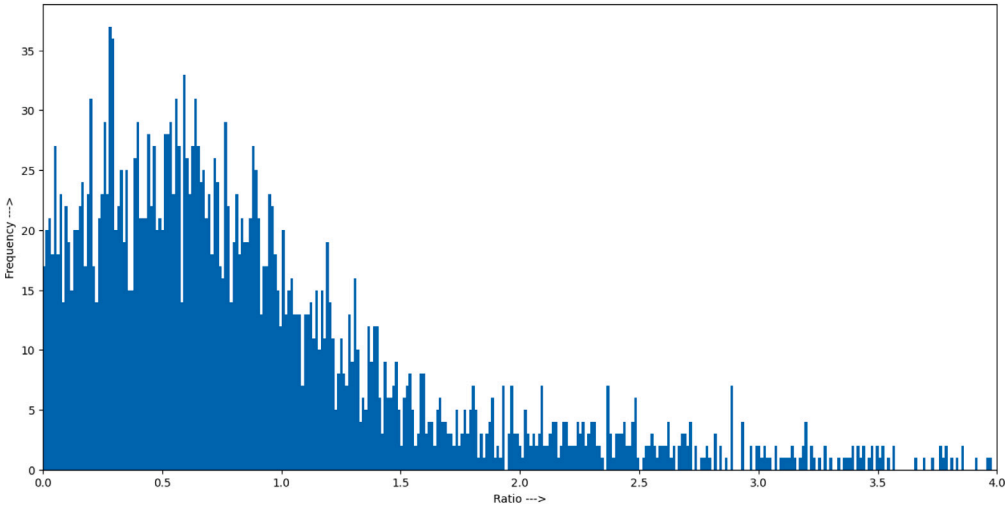


Fig. 7. Ratio of head-hip to hip-ankle distances towards Y-axis corresponding to 'No Fall' events.

of change in area for 'Fall' and 'No Fall' events corresponding to the same video are shown in Fig. 6. From this figure, it is seen that for 'No Fall' event, $\mu^A_{nf} = 52.4$ and $\sigma^A_{nf} = 1360.88$. It is also found that, for 'Fall' event, $\mu^A_f = 1152.8$ and $\sigma^A_f = 9438.35$. For the entire video having both 'Fall' and 'No Fall' scenarios, $\mu^A = 602.6$ and $\sigma^A = 5399.61$. Considering ten annotated videos, average of ten μ^A and σ^A are 600.92 and 5412.8, respectively. Therefore, for change in area, two thresholds, $Th^3_f = 3610.3$ and $Th^3_{nf} = 3610.3$ are selected.

For pose granules, head-hip ankle ratio is analyzed for the same video along both Y-axis and X-axis. The mean distribution of head-hip-ankle ratio along Y axis for 'No Fall' event is shown in Fig. 7. From this figure, it is seen that $\mu^{hha,y}_{nf} = 0.52$, $\sigma^{hha,y}_{nf} = 0.06$. The mean distribution of head-hip-ankle ratio along Y axis for 'Fall' event is shown in Fig. 8. From this figure, it is seen that $\mu^{hha,y}_f = 0.74$, $\sigma^{hha,y}_f = 0.1$. The mean distribution of head-hip-ankle ratio along X axis for 'No Fall' event is shown in Fig. 9. From this figure, it is seen that $\mu^{hha,x}_{nf} = 0.51$,

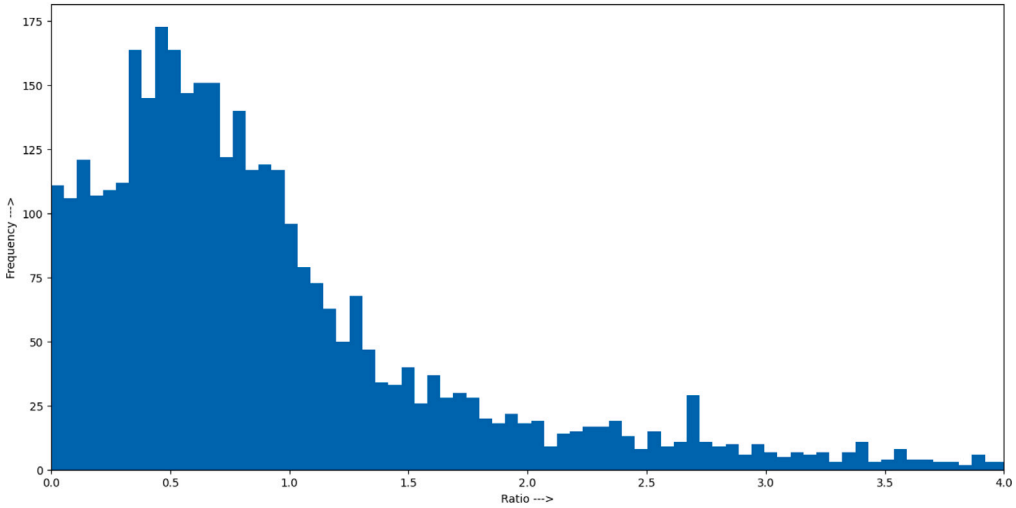


Fig. 8. Ratio of head-hip to hip-ankle distances towards Y-axis corresponding to 'Fall' events.

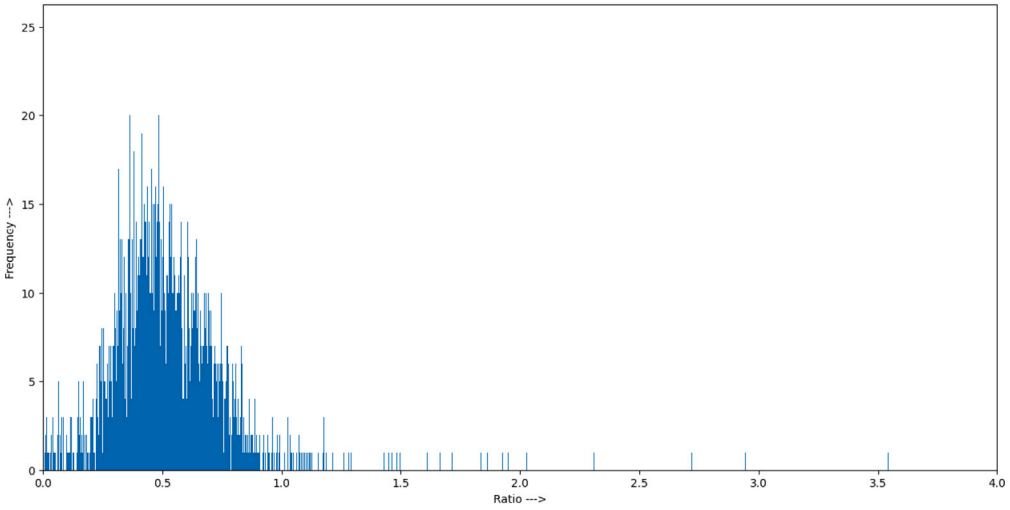


Fig. 9. Ratio of head-hip to hip-ankle distances towards X-axis corresponding to 'No Fall' events.

$\sigma_{nf}^{hha,x} = 0.04$. The mean distribution of head-hip-ankle ratio along X axis for 'Fall' event is shown in Fig. 10. From this figure, it is seen that $\mu_{f}^{hha,x} = 0.67$, $\sigma_{f}^{hha,x} = 0.09$. For the entire video having both 'Fall' and 'No Fall' scenarios, $\mu^{hha\left(=\frac{d(HH)}{d(HA)}\right)} = 0.61$ and $\sigma^{hha} = 0.065$. Considering ten annotated videos, average of ten μ^{hha} and σ^{hha} are 0.604 and 0.061, respectively. Therefore, for head-hip-ankle ratio, two thresholds, $Th_{f}^{4} = 0.665$ and $Th_{nf}^{4} = 0.543$ are selected. In this way, parametric study is done for obtaining object granules and pose granules. The parametric study for temporal self attention mechanism-based Bi-LSTM is presented in the next section.

4.2.2. Parametric study for granulated Bi-LSTM

Various parameters, namely optimizer, loss function, number of epochs, optimal frame-batch, and learning rate are used in the temporal self attention mechanism-based granulated Bi-LSTM. The optimizer and loss function used in this study are 'Adam' and 'Cross Entropy Loss', respectively. For defining the optimal number of epochs, the training losses are computed for the granulated Bi-LSTM. A graph of training losses is shown in Fig. 11. From this figure, it is seen that the minimum loss is obtained at 70 epochs. Therefore, 70 epochs are considered as the optimum in the developed TLG-LSTM. The main objective of finding the optimal frame-batch is to capture the complete fall action accurately. Frame-batch contains a number of frames. To obtain the optimal number of frames, aspect ratio of falling person is analyzed. For this experiment, two cameras are placed at both perpendicular and horizontal planes to capture the same scenario at different angles. For a Fall scenario, the progression of aspect ratios (of falling person) over frames captured by a camera that is placed at perpendicular plane, is presented in Fig. 12(a). Whereas, Fig. 12(b) presents the progression of aspect ratios over frames captured by a camera that is placed at horizontal plane. During Fall, aspect ratio of falling person gradually decreases. From both these figures, it is evident that, aspect ratio is minimum at 30th frame of a frame-batch. After 30th frame, aspect ratio slightly increases. It means either fallen person picked up or stands up. Training accuracy is also considered for obtaining the optimal frame-batch. Graphs between training accuracy and frame-batches of having different frames are presented in Fig. 13. Figs. 13(a), 13(b), 13(c), and 13(d) represents the graphs of training accuracy with 70 epochs for frame-batches having 15, 25, 30, and 35 frames, respectively. From these figures, it is evident that, maximum training accuracy is obtained for the frame-batch having 30 frames. Therefore, optimal frame-batch contains 30 frames that are required to capture the entire fall scenario.

Thirty frames that are used for fall detection tasks, are fed to the detectors (YOLOv8+Faster RCNN) for object detection. After detection, each detected object is fitted with a bounding box and used for tracking using Deep SORT algorithm. During the detection task,

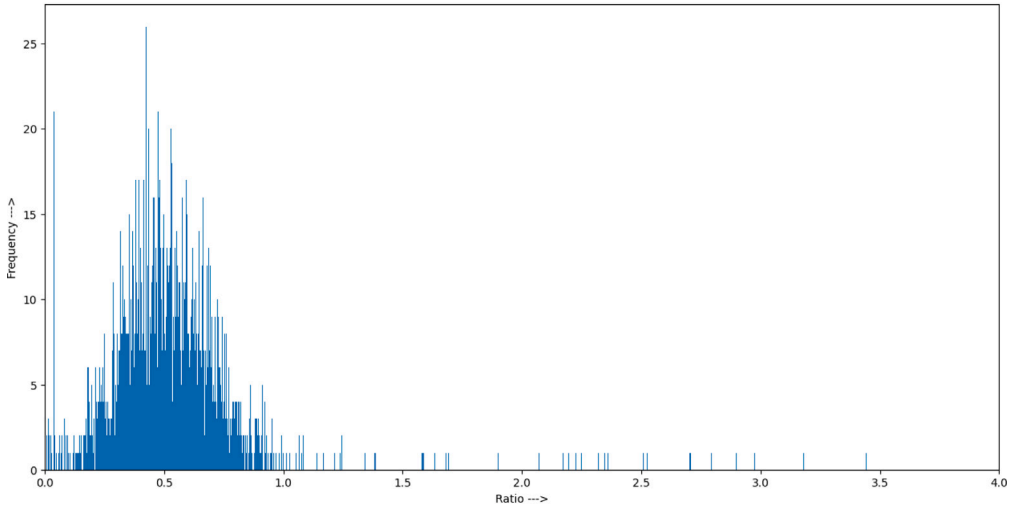


Fig. 10. Ratio of head-hip to hip-ankle distances towards X-axis corresponding to 'Fall' events.

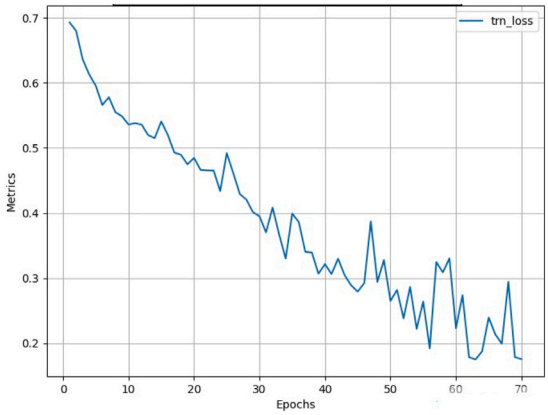


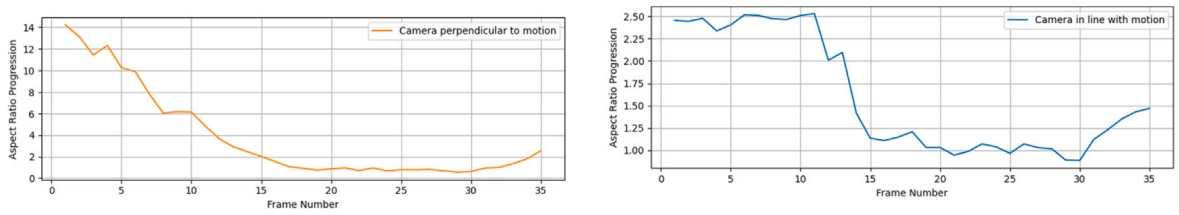
Fig. 11. Training accuracy with Epochs for granulated Bi-LSTM.

suppose any person is detected initially, but suddenly he/she is disappeared in an intermediate frame, then the bounding box information corresponding to the previous frame is considered until this person is detected again. For each detected object, eight object-level and pose features, namely change in aspect ratio, speed variation, change in area, centroid-location, head-location, hip-location, ankle-location, and head-hip-ankle ratio along X-axis and Y-axis are used for re-training of Bi-LSTM using temporal self attention mechanism-based transfer learning. Therefore, for each object detected at least once in the frame-batch, a 30×8 matrix (input size) is formed. This input is fed to the Bi-LSTM for the classification task.

For a current frame-batch, the information of previous frame-batch is preserved to reduce the information loss corresponding to an input. This is achieved by implementing a sliding window technique, where the first frame-batch contains 30 frames and the next frame-batch contains a new frame added with the 29 frames of the previous frame-batch. This minimize the information loss and reduce the computational time. Another hyper-parameter, namely learning rate is used to check how quickly a model can train. The graphs of training losses over 70 epochs corresponding to various learning rates, including $1e^{-02}$, $1e^{-03}$, $1e^{-04}$, and $1e^{-05}$ are presented in Figs. 14(a), 14(b), 14(c), 14(d), respectively. From Figs. 14(a), 14(b), it is evident that the training loss is not gradually decreased. Whereas, from Figs. 14(c), 14(d), it is evident that the training loss is gradually decreased. It means, both learning rates, $1e^{-04}$ and $1e^{-05}$ can be used for training process. To obtain the optimal learning rate, the graphs of training and validation accuracy over 70 epochs for $1e^{-04}$, and $1e^{-05}$ are presented in Figs. 15(a) and 15(b), respectively. From these figures, it is evident that, for the learning rate, $1e^{-04}$, validation accuracy is coincided with the training accuracy. Whereas, for the learning rate, $1e^{-05}$, validation accuracy does not coincide with the training accuracy. Therefore, learning rate of $1e^{-04}$ is used to obtain better model's performance. Experimental results are presented in the next section.

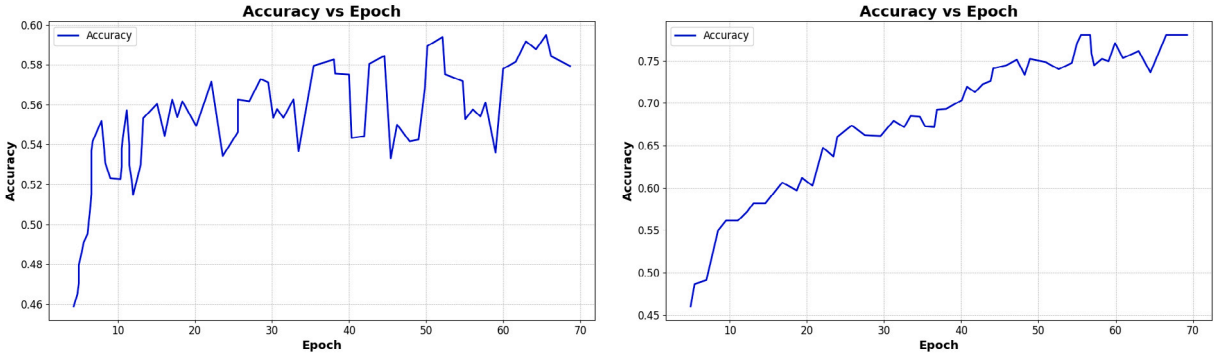
4.3. Experimental results

The effectiveness of the developed TLG-LSTM is evaluated for real-time fall detection using an extensive experiments, in line with three objectives. Objective (i) aims to demonstrate the efficacy of granulation in detecting the approximate fall location, as explained in Section 4.3.1. Objective (ii) aims to demonstrate the utility of transfer learning, as explained in Section 4.3.2. Finally, objective (iii) demonstrates the superiority of TLG-LSTM over some state-of-the-art methods, as demonstrated in Section 4.3.3.

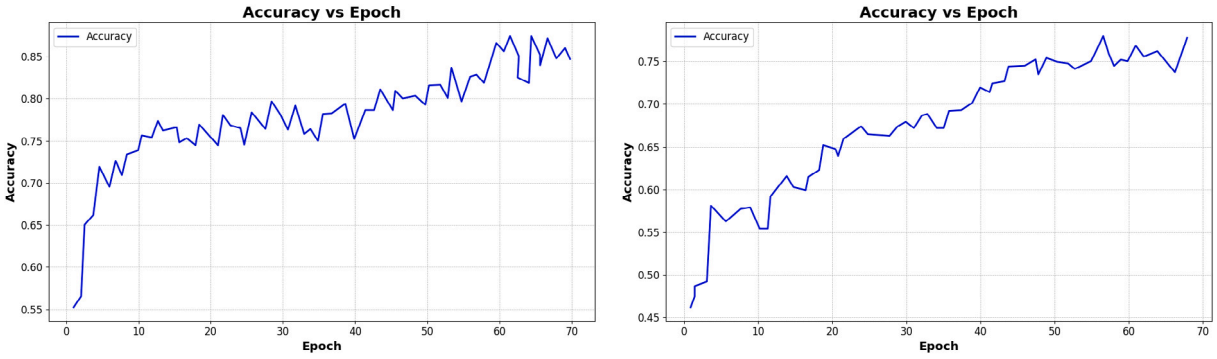


(a) Camera at perpendicular position (b) Camera at in-line position

Fig. 12. Aspect-ratio progression over Fall frames.

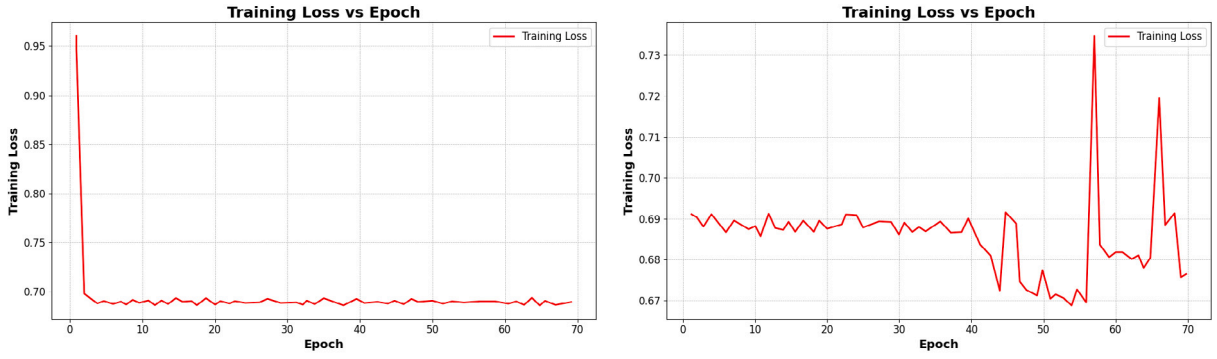


(a) Frame-batch having 15 frames (b) Frame-batch having 25 frames

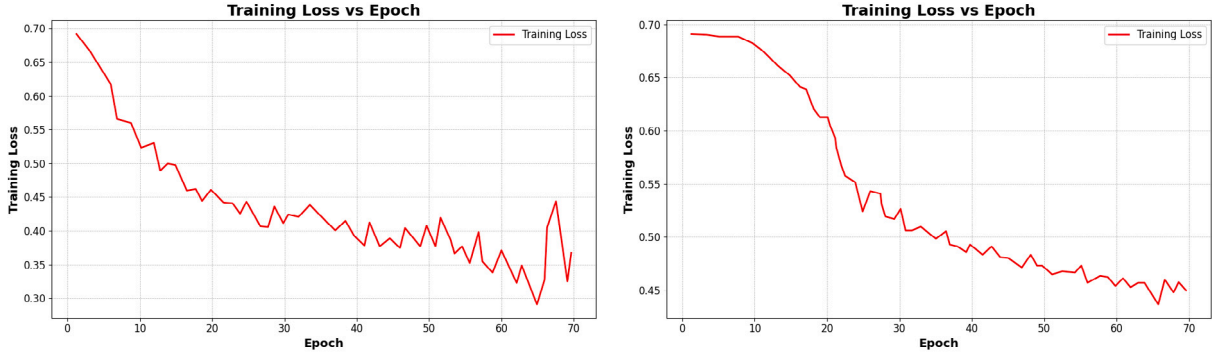


(c) Frame-batch having 30 frames (d) Frame-batch having 35 frames

Fig. 13. Training accuracy with different frame-batches.

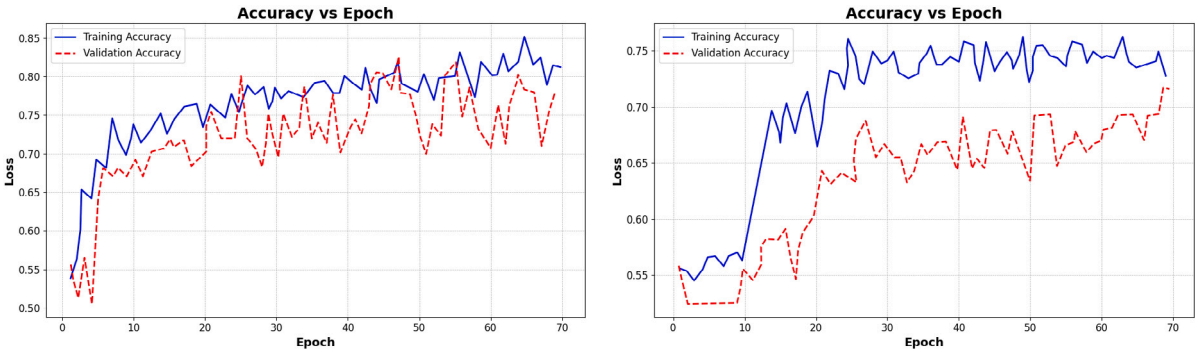


(a) Learning rate $= 1e^{-02}$ (b) Learning rate $= 1e^{-03}$



(c) Learning rate $= 1e^{-04}$ (d) Learning rate $= 1e^{-05}$

Fig. 14. Training loss with different learning rates.



(a) Learning rate = $1e^{-04}$ (b) Learning rate = $1e^{-05}$

Fig. 15. Training and validation accuracy with different Learning Rates.

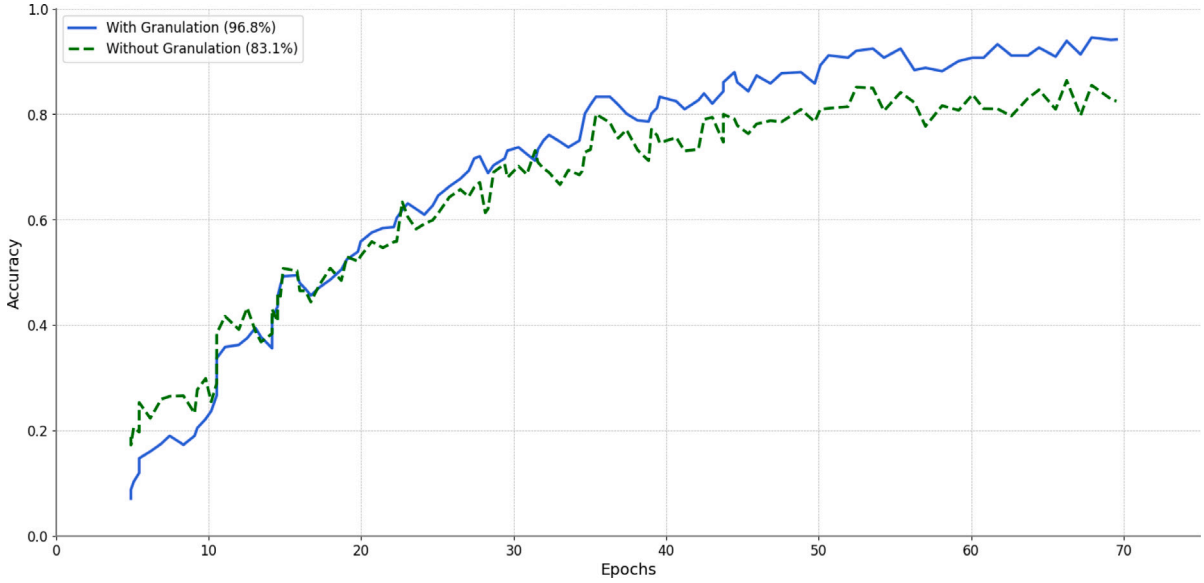


Fig. 16. Ratio of head-hip to hip-ankle distances towards X-axis corresponding to 'Fall' events.

Table 2
Comparison between traditional Bi-LSTM and granulated Bi-LSTM.

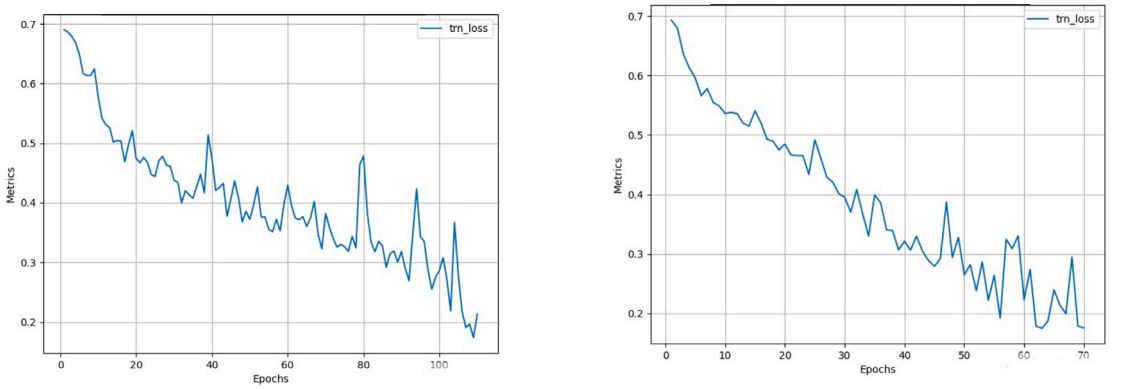
Network	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Bi-LSTM	TouTube8M	83.1	78.2	81.7	79.8
Granulated Bi-LSTM	YouTube8M	96.8	91.6	95.3	93.4

4.3.1. *Effectiveness of granulation in detecting approximate fall location*

As already defined, granulation is done to narrow the working space thereby reducing the computational complexity. Moreover, granules represent the approximate fall locations. The probability of detecting falls in those granules is greater than the probability of detecting falls in the entire frame. This increases the detection accuracy. The training accuracy of traditional Bi-LSTM (entire frame is used as input to the Bi-LSTM) and granulated Bi-LSTM (granules are used as input to the Bi-LSTM) networks over 70 epochs are presented in Fig. 16. From this figure, it is evident that training accuracy is increased for granulated Bi-LSTM. The average speed of granulated Bi-LSTM for fall detection is $16fps$ which is satisfactory speed considering the number of transfer learning blocks used in the Bi-LSTM. Four performance metrics, namely precision, recall, accuracy, and F1 score are used to evaluate the effectiveness of granulated Bi-LSTM over the traditional Bi-LSTM. The results are shown in Table 2. From this table it is evident that the granulated Bi-LSTM is superior to the traditional Bi-LSTM in terms of precision, recall, accuracy, and F1 score. All these prove the effectiveness of granulation in extracting the approximate fall locations. The effectiveness of using transfer learning for re-training of granulated Bi-LSTM is presented in the next section.

4.3.2. *Effectiveness of transfer learning*

The transfer learning is useful in leveraging the knowledge of pre-trained model that is learned on large datasets to improve the fall detection performance for small datasets. The pre-trained model is learned on large datasets having different types of labeled objects and actions. By using transfer learning, the pre-trained model is fine tuned to learn some specific features that are relevant to fall scenario. In this way, the amount of data and training time can be reduced. As already defined, temporal self attention mechanism-based transfer learning is used in Bi-LSTM for improving the speed as well as accuracy. The importance of temporal self attention mechanism in terms of training loss is presented in Fig. 17. Here, Fig. 17(a) shows the training loss over epochs for Bi-LSTM network without temporal self attention mechanism. From this figure, it is evident



(a) Without temporal self attention mechanism (b) With temporal self attention mechanism

Fig. 17. Effect of temporal self attention mechanism on Training Loss.

that, the training loss attains minimum value at 110 epochs. Whereas, Fig. 17(b) shows the training loss over epochs for Bi-LSTM network with temporal self attention mechanism. From this figure, it is evident that, the same training loss is occurred at 70 epochs. This means learning rate is faster, thereby resulting in higher speed for temporal self attention mechanism-based Bi-LSTM as compared to the without transfer learning-based Bi-LSTM network. This shows the effectiveness of temporal self attention mechanism-based transfer learning for re-training of granulated Bi-LSTM. The effectiveness of TLG-LSTM over some state-of-the-art models is discussed in the next section.

4.3.3. Comparative study

To assess the performance of TLG-LSTM for fall detection, three types of comparative studies based on: (i) fall classification, (ii) computational complexity, and (iii) robustness checking are done, as discussed in the following sections.

(i) Fall classification

A comparison is done between the developed TLG-LSTM and some state-of-the-art fall detection models to validate the superiority of TLG-LSTM. The experiments are done over some videos acquired from YoUTube8M and URFD data. Three performance metrics, namely recall, precision, and accuracy are used for this comparative study. Various well-known fall detection models, including Posture-based (PS) [54], Human skeleton-based (HS) [55], Body silhouette-based (BS) [56], Gravity-based (GV) [57], Posture and color matching-based (PCM) [58], Dynamic image-based (DI) [59], Trajectory-based (TJ) [60], Transfer learning-based (TL) [61], Deep CNN-based (DCNN) [62], Spatio-temporal graph CNN-based (STGCNN) [63], HActivityNet [46], I3D [47], and MS-TCN [48] are used for comparative study. All these models are applied over either YouTube8M or URFD dataset.

In PS and HS models, threshold-based learning strategy is used over two features, namely human posture and human skeleton, respectively. In BS-based fall detection, SVM-classification is done followed by silhouette ratio computation. Whereas, in GV model, focused time delay neural network is trained using two features, namely center of gravity and head position. In PCM model, Color matching ellipse is trained using head and ankle information. In DI model, change in posture over dynamic images is used for rule-based fall detection. In TJ model, trajectory weighted convolution features are used for fall detection. In TL model, transfer learning is used. In DCNN, deep CNN is trained over human skeleton. Whereas, in STGCNN, a graph CNN is trained using spatio-temporal features. In HActivityNet, a HAR-Net is developed for human activity recognition. Whereas, I3D is re-trained using Kinetics-Fall dataset. The temporal information of multi-scale actions is used for re-training of MS-TCN. In our developed model, TLG-LSTM, transfer learning-based granulated Bi-LSTM is used for modeling human falls.

The results of aforesaid models over URFD and YouTube8M data are exhibited in Table 3. From this table, it is evident that the developed TLG-LSTM surpasses some of the aforesaid state-of-the-art models for YouTube8M videos. It is also evident from this table that the developed TLG-LSTM is comparable with aforesaid state-of-the-art models for URFD videos. The dataset, YouTube8M contains real-life complex traffic scenarios, whereas, URFD contains occlusion-free home scenarios. Therefore, some state-of-the-art models perform good for URFD dataset but not for YouTube8M dataset. The results of various models over URFD data are presented in Rows 1 to 14 in Table 3. For most of the state-of-the-art models (refer to Rows 1, 2, 4, 5, 6, and 9 in Table 3), posture features corresponding to the previous frames are not considered. These models cannot capture the fall scenario where unusual posture (related to either picking up a coin or exercising) is found. Whereas, in some state-of-the-art models (refer to Rows 3, 7, 8, and 10 in Table 3), trajectory information is analyzed for fall detection. These models can capture fall location more accurately. Transfer learning-based models (refer to Rows 8, 9, 10, 11, 12, and 13 in Table 3) are good in fall detection. In our developed TLG-LSTM (refer to Row 14 in Table 3), both pose and trajectory-based features are used for obtaining granules that are used as input to the transfer learning-based Bi-LSTM (deep network) for fall detection. The developed TLG-LSTM provides comparable accuracy with better robustness. The results of 'TJ', 'TL', 'DCNN', 'STGCNN', and TLG-LSTM (our development)' over YouTube8M dataset are presented in Rows 15 to 19 in Table 3. From an extensive study, it is found that the developed TLG-LSTM can successfully classify 316 videos out of 345 test videos. All these results prove the supremacy of TLG-LSTM over some state-of-the-art models.

Some images of the real-time pedestrian falls near the pedestrian crossing are shown in Figs. 18 and 19. All these results are generated over two YouTube8M videos using the developed TLG-LSTM model. From Figs. 18 and 19, it is evident that two pedestrian falls are occurred near the same pedestrian crossing but at different time. The developed TLG-LSTM model successfully detect fall events on-time for taking corrective actions to mitigate the impact of fall, thereby enhancing the pedestrian safety.

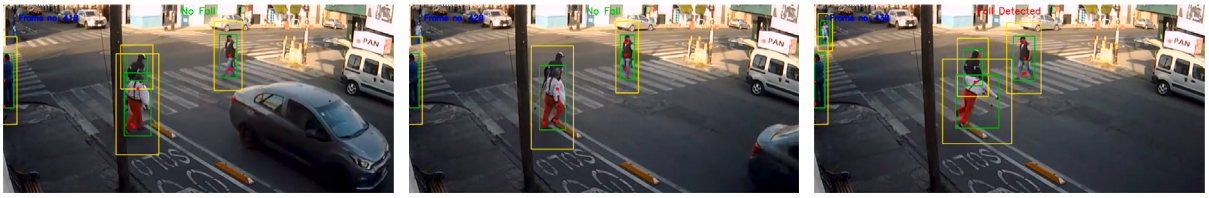
(ii) Computational complexity

The developed TLG-LSTM consists of four stages: (a) object detection and tracking, (b) feature computation, (c) granulation, and (d) temporal self attention mechanism-based Bi-LSTM for classification. The computational complexity (CoC) of each stage is explained below.

(a) CoC for object detection and tracking: Object detection and tracking are done using YOLOv8+ Faster RCNN and data association, respectively. YOLOv8 and Faster RCNN consist of ResMet50 and VGG16 networks, respectively. Let $N \times N$ be the size of the input fed to the YOLOv8 and Faster RCNN for object detection. The CoC of YOLOv8 is

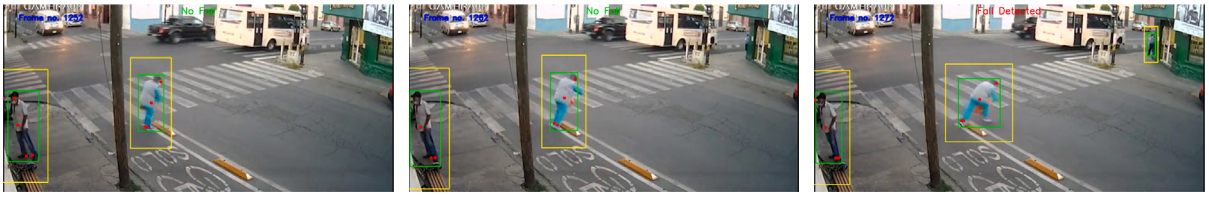
Table 3
Detection results.

Models	Learning strategy	Datasets	Recall (%)	Precision (%)	Accuracy (%)
PS [54]	Threshold-based	URFD	85.2	81.4	86.1
HS [55]	Threshold-based	URFD	89.2	85.9	90.09
BS [56]	Directed Acyclic SVM	URFD	97.2	99.2	97.08
GV [57]	Focused Time Delay NN	URFD	95.9	94.1	98.3
PCM [58]	Threshold-based	URFD	92.4	90.1	96.8
DI [59]	Temporal modeling	URFD	81.7	83.65	83.86
TJ [60]	Temporal modeling	URFD	94.7	95.70	96.04
TL [61]	Transfer Learning	URFD	93.2	94.5	97.5
DCNN [62]	Deep network	URFD	89.1	90.08	90.7
STGCNN [63]	Spatio-Temporal Graph CNN	URFD	98.7	96.5	98.6
HActivityNet [46]	Transfer learning	URFD	91.3	90.8	91.0
I3D [47]	2D ConvNet inflation	URFD	89.7	87.5	88.2
MS-TCN [48]	Transfer learning	URFD	93.2	92.4	92.9
TLG-LSTM (Ours)	Transfer Learning	URFD	98.3	96.5	98.6
TJ [60]	Temporal modeling	YouTube8M	81.2	80.6	83.2
TL [61]	Transfer Learning	YouTube8M	80.7	81.5	81.5
DCNN [62]	Deep network	YouTube8M	78.2	79.08	78.6
STGCNN [63]	Spatio-Temporal Graph CNN	YouTube8M	83.5	82.7	87.1
TLG-LSTM (Ours)	Transfer Learning	YouTube8M	95.3	86.50	91.6



(a) Frame no 418 (b) Frame no 428 (c) Frame no 438

Fig. 18. Results of pedestrian falls near the pedestrian crossing for one YouTube8M video.



(a) Frame no 1252 (b) Frame no 1262 (c) Frame no 1272

Fig. 19. Results of pedestrian falls near the pedestrian crossing for another YouTube8M video.

$O(NM)$, where M defines the number of frames present in a video. The CoC of Faster RCNN is $O(N^2M)$. The CoC for object detection ($O(det)$) is defined as:

$$O(det) = O(NM) + O(N^2M) \tag{17}$$

The CoC for tracking ($O(trk)$) task is defined as:

$$O(trk) = O\left(\frac{n \times (n-1)}{2}\right) \tag{18}$$

where n defines the number of detected persons present in a frame. For object detection and tracking, the CoC is defined as:

$$O(dettrk) = O(NM) + O(N^2M) + O\left(\frac{n \times (n-1)}{2}\right) \tag{19}$$

(b) CoC for feature computation: For each detected and tracked object, both MoveNet-Lightening and object-level features are computed. For MoveNet-Lightening feature computation, the required CoC is $O(n)$. Three object-level features, namely aspect ratio, velocity, and area are considered. The required CoC for object-level feature computation is $O(3n)$. Total CoC for feature computation is $O(4n)$.

(c) CoC for granulation: As defined earlier, two granules, namely object granule and pose granule are defined using the extracted features. The required CoC for object granule formation corresponding to a frame is $O(11n)$. Whereas, the required CoC for pose granule formation corresponding to a frame is $O(4n)$. Commonality between these two granules is used for obtaining the approximate fall location(s) more appropriately. The required CoC for obtaining this common granule is $O(2n)$. Total CoC for granulation is $O(11n) + O(4n) + O(2n) = O(17n)$.

(d) CoC for transfer learning-based Bi-LSTM: As said earlier, X represents the length of the input sequence. Due to the use of transfer learning 25% time is saved, therefore the CoC for transfer learning-based Bi-LSTM is $O\left(\frac{3X}{4}\right)$.

The total CoC for TLG-LSTM is defined as:

$$O(TLG-LSTM) = O(N^2M) + O(NM) + O\left(\frac{n \times (n-1)}{2}\right) + O(21n) + O\left(\frac{3X}{4}\right) \tag{20}$$

The CoC of a model is measured in terms of runtime/speed. The runtime of some other state-of-the-art models, including TJ, TL, DCNN, and STGCNN, and our developed TLG-LSTM are 18fps, 16fps, 11fps, 10fps, and 13fps, respectively. It is evident that the developed TLG-LSTM is superior to DCNN and STGCNN and comparable with TJ and TL. In the developed TLG-LSTM, granulation is done over the MoveNet-Lightening and object-level features for obtaining the fall location.

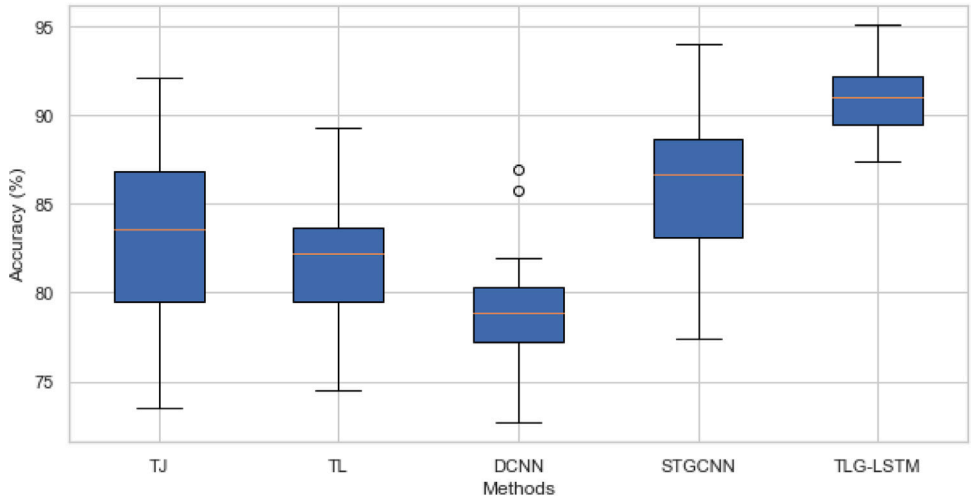


Fig. 20. Robustness checking result over YouTub8M dataset.

Accurate detection of person also affects the fall detection accuracy. Moreover, the granules (belonging to the approximate falls) are passed to the Bi-LSTM network for fall classification. Due to the aforesaid tasks, the developed TLG-LSTM takes larger time as compared to TJ and TL. It is also seen from the Table 3 that TLG-LSTM is superior to the aforesaid stat-of-the-art fall detection models in terms of precision, recall, and accuracy. In order to make a trade off between fall detection accuracy and speed, TLG-LSTM is the prime choice.

(iii) Robustness checking

A total of eighteen videos acquired from YouTube8M dataset are utilized to investigate the robustness of the TLG-LSTM for detecting human falls. Consequently, a total of eighteen accuracy are obtained which are used to generate the box-plot for robustness checking. The robustness of TLG-LSTM in terms of box plot analysis for fall detection is shown in Fig. 20. It is evident from this figure that the developed TLG-LSTM for multi-fall detection shows the higher accuracy with lowest range of dispersion as compared to some state-of-the-art models. Hence, the robustness of TLG-LSTM is proved.

5. Conclusion

Due to various road-related issues, such as potholes, pedestrian negligence, collisions, presence of abandoned objects on road, and near-misses, 'Falls' on road are increasing day-by-day. Real-time detection of human falls and prompt precautionary actions are necessary to mitigate the impact of falls. Most of the existing studies focus on simulated data that are designed for either outdoor (i.e., construction area) or indoor (i.e., home) falls. Furthermore, there is a limited research on deep learning-based models to handle the uncertainty issue that presents among various 'Fall' and 'No Fall' events corresponding to complex scenarios. These issues are handled by developing a new model, named transfer learning-based granulated Bi-LSTM (TLG-LSTM) for detecting both outdoor (i.e., road) and indoor (i.e., home) falls. To demonstrate the efficacy of the developed TLG-LSTM, comparisons with some state-of-the-art models are conducted using two datasets: YouTube8M [10] and URFD [49]. Four performance metrics, namely precision, recall, accuracy, and runtime, are used for the comparison. From the comparative study, it is found that the developed TLG-LSTM is applicable for multi-fall detection in complex scenarios, enabling its usage in a wide variety of instances. The developed TLG-LSTM also proves its superiority in terms of both speed and accuracy. The robustness of TLG-LSTM is also demonstrated in this study. Our study aims to contribute to both theoretical and practical aspects.

5.1. Theoretical contributions

For real-time fall detection, a new model, namely TLG-LSTM is developed by incorporating the concept of granulation and transfer learning with the Bi-LSTM. The developed TLG-LSTM has four parts: (i) object detection and tracking, (ii) feature generation for the detected objects, (iii) granule formation using the extracted features, and (iv) Bi-LSTM-based classification for the defined granules as 'Fall' or 'No Fall'. Various MoveNet-Lightening (e.g., head-hip distance and hip-ankle distance) and object-level features (e.g., change in aspect ratio, speed variation, and change in area) are computed for all the detected objects (person). Unlike state-of-the-art models, TLG-LSTM uses aforesaid two type of features for better modeling of both indoor and outdoor falls by generating two granules, namely pose granule and object granule. The commonality between these two granules represents the probable fall regions which are fed to the Bi-LSTM network for detecting the actual fall location. The resultant network is called granulated Bi-LSTM. A temporal self-attention mechanism is introduced to re-train the granulated Bi-LSTM. All these constitute the model, namely TLG-LSTM. Instead of using the entire frame, only granules are used for detection tasks, enabling higher accuracy and speed. Introducing the concept of granulation with the Bi-LSTM for fall detection is unique.

5.2. Practical implications

From a practical standpoint, the real-time fall detection system (FDS) can play a significant role in enhancing human safety in various environments, such as road, home, and construction site. The FDS is used for real-time monitoring through 24 × 7 live stream videos. The developed TLG-LSTM model can be deployed in the FDS for real-time fall detection. The FDS processes each frame of live stream video through the TLG-LSTM. After fall detection, corresponding video clip is generated through FDS. In future, this video clip can be analyzed to find the probable cause(s) of human falls that may be attributed to the abnormal behavior (e.g., loss of balance, and distracted walking) or environmental factors (e.g., pothole(s), obstacle(s), and slippery surface(s)). For behavior-related causes, the video clip can be reviewed to provide feedback to the concerned individual. It is done to make the person aware of his/her mistake and train the person to adopt safer behavior. This preventive approach may be useful to reduce the frequency of human falls caused by behavioral factors. If the cause is related to environmental factors (hazard(s) in road/construction site), the FDS can relay this information to the responsible authorities for taking corrective actions (e.g., repairing the hazard and/or removing

Table 4
Notations.

Notations	Meaning	Notations	Meaning
f_t	Current frame/tth frame	o_i^t	ith detected object at f_t
$y_{head_i^t}$	Location of head for o_i^t corresponding to y-axis	i	Variable
$y_{hip_i^t}$	Location of hip for o_i^t corresponding to y-axis	$y_{ankle_i^t}$	Location of ankle for o_i^t corresponding to y-axis
$x_{hip_i^t}$	Location of hip for o_i^t corresponding to x-axis	$x_{ankle_i^t}$	Location of ankle for o_i^t corresponding to x-axis
$x_{head_i^t}$	Location of head for o_i^t corresponding to x-axis	AR_i^t	Aspect ratio for o_i^t
$d(HH_i^t)$	Distance between the locations of head and hip for o_i^t	$d(HA_i^t)$	Distance between the locations of hip and ankle for o_i^t
V_i^t	Speed of o_i^t	A_i^t	Area of o_i^t
B_i^t	Bounding-box fitted over o_i^t	w_i^t	Width of B_i^t
(x_i^t, y_i^t)	Location of centroid for B_i^t	h_i^t	Height of B_i^t
Vx_i^t	Velocity component of o_i^t corresponding to x-axis	Vy_i^t	Velocity component of o_i^t corresponding to y-axis
f_{t-3}	$(t-3)^{th}$ frame	(x_i^{t-3}, y_i^{t-3})	Location of centroid for o_i^t at f_{t-3}
$\{OG_f\}$	Set of object granules corresponding to 'Fall' scenario	$\{OG_{nf}\}$	Set of object granules corresponding to 'No Fall' scenario
$d(AR_i^t)$	Change in aspect ratio for o_i^t from f_t to f_{t-1}	$d(V_i^t)$	Change in velocity for o_i^t from f_t to f_{t-1}
$d(A_i^t)$	Change in area for o_i^t from f_t to f_{t-1}	Th_f^1	Threshold used for change in aspect ratio for $\{OG_f\}$
Th_f^2	Threshold used for change in velocity for $\{OG_f\}$	Th_f^3	Threshold used for change in area for $\{OG_f\}$
Th_{nf}^1	Threshold used for change in aspect ratio for $\{OG_{nf}\}$	Th_{nf}^2	Threshold used for change in velocity for $\{OG_{nf}\}$
Th_{nf}^3	Threshold used for change in area for $\{OG_{nf}\}$	μ_{AR}	Mean of a set containing aspect ratios for an object over all frames present in a video
σ_{AR}	Standard deviation of a set containing aspect ratios for an object over all frames present in a video	P	Total number of frames in a video
$\{PG_f\}$	Set of pose granules corresponding to 'Fall' scenario	$\{PG_{nf}\}$	Set of pose granules corresponding to 'No Fall' scenario
Th_f^4	Threshold used for obtaining pose granules corresponding to the 'Fall' scenario	Th_{nf}^4	Threshold used for obtaining pose granules corresponding to the 'No Fall' scenario
$\{CG_f\}$	Set of common granules formed over $\{OG_f\}$ and $\{PG_f\}$ for 'Fall' scenarios	$\{CG_{nf}\}$	Set of common granules formed over $\{OG_{nf}\}$ and $\{PG_{nf}\}$ for 'No Fall' scenarios
H	Hidden state used in Bi-LSTM	X	Complete input sequence to Bi-LSTM
T	Transpose of a matrix	$\{Aw\}$	Set of softmax weights
$\{Cw\}$	Context vector	$\{Pv\}$	Pre-activation vector
$\{A_v\}$	Attention vector	$tanh$	Tanh function

obstacles) timely, thus preventing future falls in the same area. In addition to generating the video clip of falls, the FDS can be equipped with a real-time alert mechanism. After the detection of a fall event, the FDS can immediately generate an alarm signal to notify the concerned personnel about the emergency condition. This prompt notification ensures that help can be dispatched quickly to the fall location, minimizing the potential harm to the individual. By addressing both behavioral and environmental causes of falls and providing a real-time alert system, the developed TLG-LSTM-based FDS contributes to a comprehensive safety solution that enhances human safety in various real-world scenarios.

5.3. Effectiveness of TLG-LSTM

The key advantages of the developed TLG-LSTM model for fall detection are as follows: (i) It addresses the uncertainty issue that arises among various 'Fall' and 'No Fall' events under complex scenarios. (ii) The combination of MoveNet-Lightening features with object-level features enables TLG-LSTM for better modeling of falls using a single unified system. (iii) Incorporating the concept of granulation with the Bi-LSTM network ensures accurate detection of fall locations. (iv) Incorporating the concept of a temporal self attention mechanism within Bi-LSTM enhances the training speed and accuracy. (v) The developed TLG-LSTM model can be used to enhance human safety.

5.4. Limitations and future scopes

The present study has a few limitations. The developed TLG-LSTM model classifies the event 'a person bends down to pick up an object' as a 'Fall'. Only two MoveNet-Lightening features and three object-level features are considered. Including more features may enhance the detection accuracy. The current study is restricted to the monocular vision only. In the future, one can consider developing a fall detection model applicable to the stereo vision. In case of complex traffic scenarios, features related to a larger number of persons are analyzed for fall detection, reducing the speed. Incorporating explainable AI can solve the problem of mis-classification for fall events in complex scenarios (e.g., busy roads).

CRediT authorship contribution statement

Anima Pramanik: Writing – original draft, Validation, Methodology, Formal analysis, Conceptualization. **Soumick Sarker:** Writing – original draft, Visualization, Validation, Formal analysis, Data curation, Conceptualization. **Sobhan Sarkar:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Sankar K. Pal:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors have no relevant financial or nonfinancial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. There are no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

Acknowledgment

Prof. Sankar K. Pal acknowledges National Science Chair, Science and Engineering Research Board, Department of Science and Technology (SERB-DST), Government of India.

Appendix. Supplementary material

See Table 4.

Data availability

Data will be made available on request.

References

[1] World Health Organization, Falls, 2021, https://www.who.int/news-room/fact-sheets/detail/falls.
[2] S. Doulabi, H.M. Hassan, M.R. Ferguson, S. Razavi, A. Paez, Exploring the determinants of older adults' susceptibility to pedestrians' incidents, Accid. Anal. Prev. 155 (2021) 106100.
[3] W. Min, H. Cui, H. Rao, Z. Li, L. Yao, Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics, IEEE Access 6 (2018) 9324–9335.
[4] G.L. Santos, P.T. Endo, K.H.d.C. Monteiro, E.d.S. Rocha, I. Silva, T. Lynn, Accelerometer-based human fall detection using convolutional neural networks, Sensors 19 (7) (2019) 1644.
[5] K. Yang, C.R. Ahn, M.C. Vuran, S.S. Aria, Semi-supervised near-miss fall detection for ironworkers with a wearable inertial measurement unit, Autom. Constr. 68 (2016) 194–202.
[6] Y. Li, K. Ho, M. Popescu, A microphone array system for automatic fall detection, IEEE Trans. Biomed. Eng. 59 (5) (2012) 1291–1301.
[7] M.G. Amin, Y.D. Zhang, F. Ahmad, K.D. Ho, Radar signal processing for elderly fall detection: The future for in-home monitoring, IEEE Signal Process. Mag. 33 (2) (2016) 71–80.
[8] I. Boudouane, A. Makhlouf, M.A. Harkat, M.Z. Hammouche, N. Saadia, A. Ramdane Cherif, Fall detection system with portable camera, J. Ambient Intell. Humaniz. Comput. 11 (2020) 2647–2659.
[9] S.K. Pal, A. Pramanik, J. Maiti, P. Mitra, Deep learning in multi-object detection and tracking: state of the art, Appl. Intell. 51 (2021) 6400–6429.
[10] A. Pramanik, S. Sarkar, S.K. Pal, Video surveillance-based fall detection system using object-level feature thresholding and Z- numbers, Knowl.-Based Syst. 280 (2023) 110992.
[11] A. Pramanik, S.K. Pal, J. Maiti, P. Mitra, Granulated RCNN and multi-class deep sort for multi-object detection and tracking, IEEE Trans. Emerg. Top. Comput. Intell. 6 (1) (2021) 171–181.
[12] A. Núñez-Marcos, G. Azkune, I. Arganda-Carreras, et al., Vision-based fall detection with convolutional neural networks, Wirel. Commun. Mob. Comput. 2017 (2017).
[13] D. Berardini, S. Moccia, L. Migliorelli, I. Pacifici, P. di Massimo, M. Paolanti, E. Frontoni, Fall detection for elderly-people monitoring using learned features and recurrent neural networks, Exp. Results 1 (2020) e7.
[14] Y.M. Galvão, L. Portela, P. Barros, R.A. de Araújo Fagundes, B.J. Fernandes, OneFall-GAN: A one-class GAN framework applied to fall detection, Eng. Sci. Technol. Int. J. 35 (2022) 101227.
[15] G. Sun, Z. Wang, Fall detection algorithm for the elderly based on human posture estimation, in: 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC, IEEE, 2020, pp. 172–176.
[16] N. Lu, Y. Wu, L. Feng, J. Song, Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data, IEEE J. Biomed. Health Inf. 23 (1) (2018) 314–323.
[17] C.-B. Lin, Z. Dong, W.-K. Kuan, Y.-F. Huang, A framework for fall detection based on OpenPose skeleton and LSTM/GRU models, Appl. Sci. 11 (1) (2020) 329.
[18] S. Li, H. Xiong, X. Diao, Pre-impact fall detection using 3D convolutional neural network, in: 2019 IEEE 16th International Conference on Rehabilitation Robotics, ICORR, IEEE, 2019, pp. 1173–1178.
[19] S.K. Pal, B.U. Shankar, P. Mitra, Granular computing, rough entropy and object extraction, Pattern Recognit. Lett. 26 (16) (2005) 2509–2517.
[20] A. Pathak, S.K. Pal, Fuzzy grammars in syntactic recognition of skeletal maturity from X-rays, IEEE Trans. Syst. Man Cybern. 16 (5) (1986) 657–667.
[21] S.K. Pal, A. Ghosh, Fuzzy geometry in image analysis, Fuzzy Sets and Systems 48 (1) (1992) 23–40.
[22] S.K. Pal, Soft data mining, computational theory of perceptions, and rough-fuzzy approach, Inform. Sci. 163 (1–3) (2004) 5–12.
[23] A. Butt, S. Narejo, M.R. Anjum, M.U. Yonus, M. Memon, A.A. Samejo, Fall detection using LSTM and transfer learning, Wirel. Pers. Commun. 126 (2) (2022) 1733–1750.
[24] M. Laurer, W. Van Atteveldt, A. Casas, K. Welbers, Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and BERT-NLI, Political Anal. 32 (1) (2024) 84–100.
[25] J. Terven, D. Cordova-Esparza, A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond, 2023, arXiv preprint arXiv:2304.00501.
[26] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[27] D. Razum, G. Seketa, J. Vugrin, I. Lackovic, Optimal threshold selection for threshold-based fall detection algorithms with multiple features, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, IEEE, 2018, pp. 1513–1516.
[28] J. Gutiérrez, V. Rodríguez, S. Martin, Comprehensive review of vision-based fall detection systems, Sensors 21 (3) (2021) 947.
[29] W. Chen, Z. Jiang, H. Guo, X. Ni, Fall detection based on key points of human-skeleton using openpose, Symmetry 12 (5) (2020) 744.
[30] A. Osigbesan, S. Barrat, H. Singh, D. Xia, S. Singh, Y. Xing, W. Guo, A. Tsourdos, Vision-based fall detection in aircraft maintenance environment with pose estimation, in: 2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI, IEEE, 2022, pp. 1–6.
[31] E. Alam, A. Sufian, P. Dutta, M. Leo, Real-time human fall detection using a lightweight pose estimation technique, in: International Conference on Computational Intelligence in Communications and Business Analytics, Springer, 2023, pp. 30–40.
[32] A. Yajai, S. Rasmequan, Adaptive directional bounding box from RGB-D information for improving fall detection, J. Vis. Commun. Image Represent. 49 (2017) 257–273.
[33] L. Yao, W. Min, K. Lu, A new approach to fall detection based on the human torso motion model, Appl. Sci. 7 (10) (2017) 993.
[34] M.A. Mousse, C. Motamed, E.C. Ezin, Video-based people fall detection via homography mapping of foreground polygons from overlapping cameras, in: 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems, SITIS, IEEE, 2015, pp. 164–169.
[35] M.A. Mousse, B. Atohoun, Saliency based human fall detection in smart home environments using posture recognition, Heal. Inform. J. 27 (3) (2021) 14604582211030954.
[36] B. Wan, D. Zhou, Y. Liu, R. Li, X. He, Pose-aware multi-level feature network for human object interaction detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9469–9478.
[37] M.E.N. Gomes, D. Macêdo, C. Zanchettin, P.S.G. de Mattos-Neto, A. Oliveira, Multi-human fall detection and localization in videos, Comput. Vis. Image Underst. 220 (2022) 103442.
[38] S.K. Yadav, A. Luthra, K. Tiwari, H.M. Pandey, S.A. Akbar, ARFDNet: An efficient activity recognition & fall detection system using latent feature pooling, Knowl.-Based Syst. 239 (2022) 107948.
[39] M. Taufeeque, S. Koita, N. Spicher, T.M. Deserno, Multi-camera, multi-person, and real-time fall detection using long short term memory, in: Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications, 11601, International Society for Optics and Photonics, 2021, 1160109.
[40] S. Tateno, F. Meng, R. Qian, Y. Hachiya, Privacy-preserved fall detection method with three-dimensional convolutional neural network using low-resolution infrared array sensor, Sensors 20 (20) (2020) 5957.
[41] E. Alam, A. Sufian, P. Dutta, M. Leo, Vision-based human fall detection systems using deep learning: A review, Comput. Biol. Med. (2022) 105626.
[42] M.M. Islam, O. Tayan, M.R. Islam, M.S. Islam, S. Nooruddin, M.N. Kabir, M.R. Islam, Deep learning based systems developed for fall detection: A review, IEEE Access 8 (2020) 166117–166137.
[43] J. Nogas, S.S. Khan, A. Mihailidis, Deepfall: non-invasive fall detection with deep spatio-temporal convolutional autoencoders, J. Healthc. Inform. Res. 4 (1) (2020) 50–70.
[44] N. Maray, A.H. Ngu, J. Ni, M. Debnath, L. Wang, Transfer learning on small datasets for improved fall detection, Sensors 23 (3) (2023) 1105.
[45] A.N. Patel, R. Murugan, P.K.R. Maddikunta, G. Yenduri, R.H. Jhaveri, Y. Zhu, T.R. Gadekallu, AI-powered trustable and explainable fall detection system using transfer learning, Image Vis. Comput. 149 (2024) 105164.

[46] M. Khaliluzzaman, M.A.B.S. Sayem, L. KaderMisbah, HActivityNet: A deep convolutional neural network for human activity recognition, EMITTER Int. J. Eng. Technol. 9 (2) (2021) 357–376.
[47] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.
[48] Y.A. Farha, J. Gall, MS-TCN: Multi-stage temporal convolutional network for action segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3575–3584.
[49] M. Kepski, UR fall detection dataset, 2022, http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html. (Accessed 19 June 2022).
[50] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, F. Fioranelli, Bi-LSTM network for multimodal continuous human activity recognition and fall detection, IEEE Sens. J. 20 (3) (2019) 1191–1201.
[51] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft coco captions: Data collection and evaluation server, 2015, arXiv preprint arXiv:1504.00325.
[52] D. Reis, J. Kupec, J. Hong, A. Daoudi, Real-time flying object detection with YOLOv8, 2023, arXiv preprint arXiv:2305.09972.
[53] F. Yang, X. Zhang, B. Liu, Video object tracking based on YOLOv7 and DeepSORT, 2022, arXiv preprint arXiv:2207.12202.
[54] M. Yu, Y. Yu, A. Rhuma, S.M.R. Naqvi, L. Wang, 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 J. Biomed. Health Inf. 17 (6) (2013) 1002–1014.
[55] Y.-T. Chen, Y.-C. Lin, W.-H. Fang, A hybrid human fall detection scheme, in: 2010 IEEE International Conference on Image Processing, IEEE, 2010, pp. 3485–3488.
[56] M. Yu, A. Rhuma, S.M. Naqvi, L. Wang, J. Chambers, A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment, IEEE Trans. Inf. Technol. Biomed. 16 (6) (2012) 1274–1286.
[57] M. Humenberger, S. Schraml, C. Sulzbachner, A.N. Belbachir, A. Srp, F. Vajda, Embedded fall detection with a neural network and bio-inspired stereo vision, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2012, pp. 60–67.
[58] M. Shoaib, R. Dragon, J. Ostermann, View-invariant fall detection for elderly in real home environment, in: 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, IEEE, 2010, pp. 52–57.
[59] Y. Fan, M.D. Levine, G. Wen, S. Qiu, A deep neural network for real-time detection of falling humans in naturally occurring scenes, Neurocomputing 260 (2017) 43–58.
[60] Z. Zhang, X. Ma, H. Wu, Y. Li, Fall detection in videos with trajectory-weighted deep-convolutional rank-pooling descriptor, IEEE Access 7 (2018) 4135–4144.
[61] G.K. Hader, M.M. Ben Ismail, O. Bchir, Automatic fall detection using region-based convolutional neural network, Int. J. Inj. Control Saf. Promot. 27 (4) (2020) 546–557.
[62] U. Asif, B. Mashford, S. Von Cavallar, S. Yohanandan, S. Roy, J. Tang, S. Harrer, Privacy preserving human fall detection using video data, in: Machine Learning for Health Workshop, PMLR, 2020, pp. 39–51.
[63] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, Q. Dai, STAT: Spatial-temporal attention mechanism for video captioning, IEEE Trans. Multimed. 22 (1) (2019) 229–241.