International Journal of Ethical AI Application
eISSN: 3093-7736
https://ijeaa.cultechpub.com/index.php/ijeaa

A Hybrid Framework for Temporal Object Behavior Analysis Using LSTM and Real-Time Detection

Subhan Uddin[1], Babar Hussain[2], Noman Ahmad[3], Adil Hussain[4], Sidra Fareed[5]

School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China

Abstract

Fall detection in surveillance videos is a critical task with widespread applications in healthcare and public safety. In this paper, we propose a hybrid framework that combines YOLOv8 for real-time object detection with an LSTM based temporal reasoning module to classify human activities across video sequences. Our method captures both spatial appearance and temporal motion patterns, allowing it to distinguish subtle differences between activities such as standing, walking, lying, and falling. We evaluate our model on the UR Fall Detection (URFD) dataset and achieve a validation accuracy of 92.0% and a mean Average Precision (mAP@0.5) of 91.2%. Qualitative results further demonstrate robust predictions even in challenging scenarios with occlusions and lighting variations. An ablation study confirms that integrating temporal reasoning significantly boosts performance over frame-based detection alone. The proposed approach offers a promising solution for real-time fall detection in intelligent surveillance and assistive monitoring systems.

Keywords

Object Detection, Temporal Reasoning, YOLOv8, LSTM, Behavior Analysis, Video Surveillance, Human Activity Recognition

1. Introduction

Real-time object detection has significantly advanced in recent years, with models like YOLOv8 [1] achieving high accuracy and low latency across diverse applications including surveillance, autonomous driving, and robotics. These detectors are highly optimized for spatial understanding, yet most operate in a frame-by-frame manner, treating each image independently. This design, while efficient, neglects the temporal continuity inherent in video streams, limiting their ability to capture and reason about dynamic object behavior over time.

In practical scenarios such as elderly monitoring, public safety, and intelligent transportation, understanding the temporal evolution of object states is crucial. For instance, detecting a person falling or a vehicle stopping abruptly cannot be accurately inferred from a single frame—it requires recognizing patterns over multiple time steps. Traditional object detection pipelines fall short in such contexts due to their lack of temporal modeling capabilities [2,3].

To bridge this gap, we propose a hybrid framework that augments real-time object detection with temporal reasoning using sequence modeling. Specifically, we employ YOLOv8 as the spatial backbone for object detection and tracking, and an LSTM-based module to model temporal dependencies in object behavior. By feeding object-level features such as class labels, bounding box coordinates, and motion vectors over time into the LSTM, our system can infer high-level behavioral states like “walking”, “falling”, or “stopped”.

This integration of spatial and temporal modules enables our framework to detect complex behavior transitions that static models miss, offering improved performance in video-based understanding. Compared to prior work that uses either action recognition [4,5] or video object detection [6,7], our approach provides a lightweight and realtime solution suitable for deployment in edge environments.

We evaluate our system on benchmark datasets that include labeled human activity sequences and demonstrate superior performance in both object detection and temporal event recognition. Our results validate the efficacy of combining YOLOv8 with LSTM for event-aware vision tasks and open new avenues for behavior understanding in dynamic scenes.

2. Related Work

2.1 Object Detection

Object detection has seen significant advancements due to deep learning, with architectures evolving from two-stage detectors to more efficient single-stage models. Faster RCNN [8] pioneered the use of region proposal networks to simultaneously predict object locations and classes, achieving high accuracy but at the cost of slower inference speeds unsuitable for real-time applications. To address this, single-stage detectors such as SSD [9] and YOLO [1,10] were introduced, offering a favorable trade-off between speed and accuracy. In particular, YOLOv8 [1] has pushed the frontier of real-time detection with lightweight architectures and optimized training strategies, enabling deployment in resourceconstrained settings.

More recently, transformer-based object detectors like DETR [11] have leveraged self-attention mechanisms to model global context, enabling end-to-end detection without hand-designed components like anchor boxes. Variants such as RT-DETR [12] have improved inference speed, making transformers more practical for real-time use. Despite these advances, the majority of object detectors operate on a per-frame basis, treating each image independently without incorporating temporal information, which limits their ability to understand dynamic scenes or evolving object states in videos.

2.2 Action Recognition and Temporal Modeling

Temporal understanding in videos is essential for tasks such as action recognition and event detection. Early works incorporated recurrent neural networks like LSTM and GRU to capture temporal dependencies from sequential visual features [13,14]. These models demonstrated the ability to learn long-range temporal patterns, crucial for distinguishing actions spanning multiple frames.

Parallel to RNNs, convolutional neural network architectures evolved to model spatiotemporal features directly. The SlowFast network [5] introduced dual pathways processing video at different frame rates to capture both detailed spatial semantics and motion information. Meanwhile, transformer-based architectures such as TimeSformer [7] and ViViT [**?**] applied self-attention to video data, modeling global spatial-temporal interactions efficiently and achieving state-of-the-art performance in action classification.

However, these approaches primarily focus on videolevel classification and do not explicitly localize objects or their behavioral states within the scene, limiting their utility in applications requiring fine-grained event detection at the object level.

2.3 Spatio-Temporal Reasoning

Recognizing complex object behaviors requires combining accurate detection with temporal reasoning. Multi-object tracking (MOT) methods such as SORT [15] and ByteTrack [16] extend object detectors to associate instances across frames, facilitating temporal consistency. Building on tracking, some frameworks integrate temporal modules—like LSTM or transformers—to interpret sequences of detections for action or event recognition [17,18].

Despite these advancements, existing approaches often trade-off real-time performance for richer temporal context or rely on computationally intensive models that are impractical for edge devices. Additionally, few models jointly optimize spatial detection and temporal event recognition in a unified architecture. Our proposed hybrid framework addresses these challenges by coupling the real-time detection capability of YOLOv8 with an LSTM-based temporal reasoning module, enabling efficient, accurate, and eventaware object behavior analysis suited for safety-critical applications.

3. Problem Formulation

Given the increasing demand for intelligent video analysis, our work addresses the problem of *event-aware temporal object behavior analysis* in continuous video streams. Formally, the input to our system is a sequence of video frames denoted as $\mathrm{I}=\{I_t\}_{t=1}^{T}$, where T is the number of frames in a given temporal window. Alternatively, the input can be expressed as a sequence of object detection outputs generated by a frame-wise detector such as YOLOv8 [1]. Each detection at time t is represented as a set of bounding boxes and associated class probabilities:

$$D_t = \{(b_{t,i}, c_{t,i}, s_{t,i})\}_{i=1}^{N_t}$$

where $b_{t,i} \in \mathrm{R}^4$ denotes the bounding box coordinates of the i-th detected object, $c_{t,i}$ the predicted class label, $s_{t,i}$ the confidence score, and N_t is the number of detections in frame t .

The goal is to assign each detected object not only a spatial label but also a *temporal state* representing its behavior over time. More concretely, the output at each time step is defined as:

$$O_t = \{(b_{t,i}, c_{t,i}, s_{t,i}, e_{t,i})\}_{i=1}^{M_t}$$

where $e_{t,i} \in \mathrm{E}$ is the event-aware temporal state associated with the object, such as *walking*, *falling*, or *fallen*. The number of temporally tracked objects M_t may differ from N_t due to tracking and temporal association processes.

This problem formulation extends classical object detection by incorporating *event-awareness*, requiring the model to capture not only the existence and location of objects but also the *state transitions* that occur over time. The temporal state transitions encode rich semantic information about the dynamic behavior of objects, enabling applications like elderly fall detection [19], anomaly recognition in surveillance [20], and human activity understanding [13].

A key challenge arises from learning meaningful temporal dependencies between sequential detections. This involves reasoning about how object states evolve, despite potential issues such as occlusions, missed detections, and noisy bounding box predictions [15]. Recurrent architectures such as LSTMs [21] are well-suited to model these sequential dependencies, as they can maintain memory of previous states and effectively smooth noisy inputs. Moreover, integrating spatial features from detectors with temporal modeling helps the system disambiguate transient or ambiguous behaviors [22, 23].

In summary, our problem formulation can be viewed as a joint spatiotemporal learning task where the objective is to map a sequence of video frames or detection outputs to a sequence of object-level spatiotemporal states, enabling accurate and timely recognition of dynamic behaviors in realworld video streams.

4. Methodology

Our goal is to develop a real-time, event-aware object behavior analysis system that can detect, track, and temporally classify object actions across frames. The system is composed of three major modules: (1) Frame-wise object detection, (2) Inter-frame object tracking, and (3) Temporal behavior recognition. Figure 1 illustrates the complete pipeline.

4.1 System Architecture Overview

4.1.1 Stage 1: Frame-level Object Detection

We begin with frame-wise object detection using YOLOv8 [1], a modern and lightweight single-stage detector optimized for real-time applications. Each input video frame I_t is processed independently to extract bounding boxes $b_{t,i}$, predicted class labels $c_{t,i}$, and confidence scores $s_{t,i}$ for every object *i* detected at time *t:*

$$D_t = \{(b_{t,i}, c_{t,i}, s_{t,i}) \big| i = 1, \dots, N_t\}$$

YOLOv8 is selected for its balance between inference speed and accuracy, and its ability to generalize across diverse object classes and motion patterns in unconstrained environments.

4.1.2 Stage 2: Object Tracking Across Frames

The raw detections from YOLOv8 are limited to the current frame and do not capture temporal coherence. To construct meaningful object trajectories, we use a multi-object tracker. In our system, we employ ByteTrack [16] due to its superior performance in crowded and occluded scenes and its ability to associate low-confidence detections across frames. ByteTrack assigns a consistent ID to each object by combining motion prediction (via Kalman filtering) and appearance embedding matching. It outputs a set of temporally consistent trajectories:

$$T_j = \{(b_{t,j}, c_{t,j}) \big| t = t_0^j, \dots, t_n^j\}$$

Each object track T_j consists of the bounding box and class label sequence over time, providing the temporal foundation for behavior analysis.

4.2 Proposed Framework Architecture

To provide a comprehensive overview of our hybrid system, Figure 1 presents the complete architecture of the proposed framework. The system is composed of three main modules: (1) YOLOv8-based frame-level object detection, (2) ByteTrack-based multi-object tracking, and (3) an LSTMbased temporal behavior modeling module.

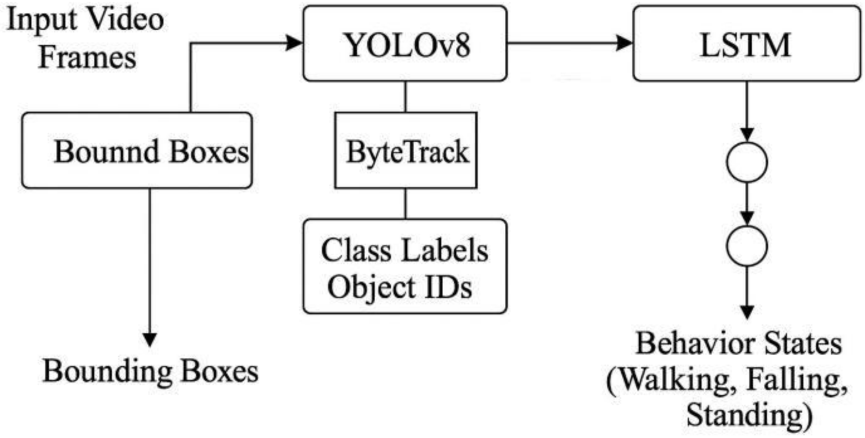

Figure 1. Overview of the proposed hybrid framework for temporal object behavior analysis.

The pipeline consists of YOLOv8 for real-time object detection, ByteTrack for inter-frame object association, and LSTM for temporal behavior recognition.

As illustrated in Figure 1, each input video frame is first processed by YOLOv8 to detect objects and extract spatial features. ByteTrack then associates detected objects across consecutive frames, generating consistent object trajectories. These trajectories, along with their spatial and motion features, are fed into the LSTM module, which outputs the predicted behavioral state for each object at each time step. This modular design enables robust, real-time analysis of complex object behaviors in video streams.

Stage 3: Temporal Behavior Modeling

While object tracking provides movement trajectories, understanding behavioral events (e.g., walking, falling, stopped) requires temporal reasoning over sequences. To achieve this, we use a Long Short-Term Memory (LSTM) network [21], a recurrent model capable of modeling longterm dependencies and handling noisy sequential data [24].

For each object trajectory T_j , we extract a sequence of features $f_{t,j}$ over a fixed time window of k frames. The LSTM consumes this temporal sequence and outputs perframe predictions for the object's behavioral state:

$$e_{t,j} = \text{LSTM}(f_{t\text{-}k+1}, j, ..., f_{t,j})$$

The output $e_{t,j}$ represents one of the predefined event labels such as "moving", "falling", or "fallen". The LSTM is trained to model both continuous actions and abrupt transitions (e.g., from "walking" to "falling") by learning contextual changes in object dynamics.

4.3 Feature Representation

Effective temporal modeling depends on robust and discriminative features that encode both spatial configuration and motion. For each object j at time t, we construct a compact feature embedding $f_{t,j}$ combining the following components:

Spatial features: normalized bounding box coordinates (x, y, w, h) , YOLOv8 class logits $c_{t,j}$, and detection confidence scores $s_{t,j}$. **Motion features**: temporal displacement $\Delta x = x_t - x_{t-1}$ and $\Delta y = y_t - y_{t-1}$, velocity magnitude, and optionally optical flow features computed from adjacent frames using lightweight flow networks such as FlowNet2 [25]. The final feature vector per timestep is defined as:

$$f_{t,j} = [x_{t,j}, y_{t,j}, w_{t,j}, h_{t,j}, c_{t,j}, s_{t,j}, \Delta x, \Delta y, v_{t,j}, \text{Flow}_{t,j}]$$

These features provide rich context for temporal modeling, allowing the LSTM to differentiate between subtle state changes like "standing" vs. "stopped" or "walking" vs. "falling".

4.4 Training Objective

To train our hybrid framework, we use a multi-task loss formulation that combines the standard object detection loss from YOLOv8 with a temporal classification loss for the LSTM. The total loss is defined as:

$$L_{\text{total}} = \lambda_1 \cdot L_{\text{det}} + \lambda_2 \cdot L_{\text{temp}}$$

L_{det} includes classification, objectness, and bounding box regression losses as in YOLOv8 [26]. L_{temp} is the cross-entropy loss between the predicted and ground-truth temporal states for each timestep.

The weights λ_1 and λ_2 control the balance between accurate spatial detection and temporal behavior learning. In practice, we set $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$ after tuning on the validation set.

4.5 Implementation Details

We implemented our system using the PyTorch deep learning framework. For object detection, we use the official Ultralytics YOLOv8 API [1], while temporal modeling is implemented using PyTorch's native ‘nn.LSTM‘ module. ByteTrack [16] is used for real-time object tracking with a re-identification threshold set to 0.7.

All experiments were conducted on a workstation with an NVIDIA RTX 3090 GPU (24GB), 128 GB RAM, and an Intel i9 CPU. The LSTM model was trained with a sequence length of 16 frames, a batch size of 32, and the Adam optimizer with a learning rate of 1×10^{-4}. We also tested deployment feasibility on Jetson Orin and NVIDIA RTX 4060 laptops to evaluate real-time performance in embedded settings.

5. Dataset and Experimental Setup

To validate the performance of our proposed hybrid framework for temporal object behavior analysis, we conduct experiments on both public action recognition datasets and domain-specific fall detection datasets. Our evaluation protocol encompasses detection, tracking, and temporal event recognition components, each assessed through standard metrics.

5.1 Datasets

5.1.1 Public Benchmarks

We utilize three widely used datasets for evaluating videobased action recognition and behavior modeling:

UCF101 [27]: This dataset consists of 13,320 video clips spread across 101 human action categories such as walking, running, jumping, and falling. While primarily designed for video-level action classification, we repurpose it by extracting frame-wise object annotations using a pretrained detector and annotating temporal transitions where applicable.

HMDB51 [28]: Containing over 7,000 clips across 51 action classes, HMDB51 provides a diverse set of scenes and motions. It presents additional challenges due to background clutter and camera motion, making it a valuable testbed for generalization and robustness.

ActivityNet [29]: This large-scale dataset includes 20,000 videos annotated with start and end timestamps for temporal actions across 200 categories. We use a subset of activities that can be tracked spatially (e.g., “sit down”, “get up”) for evaluating event-aware detection.

UR Fall Detection Dataset [30]: For domain-specific evaluation, we use the UR Fall Detection Dataset, which includes RGB-D recordings of subjects simulating activities of daily living (ADLs) and fall scenarios. It contains accurate temporal annotations for activities such as walking, sitting, falling, and lying down, making it ideal for evaluating our system’s ability to detect state transitions.

5.1.2 Data Preprocessing

All video clips are first decomposed into individual frames at 25 FPS. For datasets without bounding box annotations, we generate pseudo-labels using a pretrained YOLOv8 detector. The resulting frame-wise annotations are refined manually for fall sequences to ensure accurate trajectory supervision.

For temporal labeling, we annotate each object’s behavioral state at each frame using a finite state sequence (e.g., *walking → falling → fallen*). These labels are aligned with object tracks to train the LSTM module. We discard segments with occlusions or ambiguous motion patterns to maintain label quality.

5.2 Evaluation Metrics

We evaluate each component of the pipeline using domainappropriate metrics.

5.2.1 Object Detection

For the detection module (YOLOv8), we report the standard mean Average Precision (mAP) at multiple Intersectionover-Union (IoU) thresholds, following the COCO evaluation protocol [31]. We compute: mAP@[.5:.95] and AP_{50}, AP_{75}

5.2.2.Object Tracking

To assess the object tracking component (ByteTrack), we use standard Multiple Object Tracking Accuracy (MOTA) and the number of ID switches (ID-SW) [32]. These metrics reflect the consistency of tracking across frames:

$$\mathrm{MOTA}=1-\frac{\mathrm{FN}+\mathrm{FP}+\mathrm{ID\text{-}SW}}{\mathrm{GT}}$$

5.2.3 Temporal Event Prediction

For the temporal classification task (LSTM), we evaluate frame-wise prediction performance using Precision, Recall, and F1-score per behavioral class. This provides a finegrained view of how well the system detects transitions such as “falling” or “stopped”:

$$\mathrm{F1\text{-}score}=2\cdot\frac{\mathrm{Precision}\cdot\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}$$

We compute both macro-averaged and per-class scores to highlight performance across minority events like falls.

5.3 Baselines

To assess the benefit of our spatio-temporal integration, we compare our approach against three competitive baselines:

1. Frame-level Detection + Majority Voting: A naive baseline where YOLOv8 detects objects, and their temporal state is inferred by applying a majority vote over class predictions from each frame independently.

2. Action Recognition on Raw Frames: We implement an LSTM trained directly on raw frame-level CNN features (ResNet-50 [33]) without tracking or spatial localization. This reflects standard video classification pipelines and helps evaluate the impact of integrating object-level tracking.

3. Transformer-based Event Classifier: We compare against a temporal transformer (e.g., TimeSformer [7]) applied to sequences of global frame features. While transformers offer powerful sequence modeling, their resource consumption and lack of localized reasoning limit real-time feasibility.

Our experiments show that our proposed hybrid system significantly improves the temporal resolution and accuracy of object-level event understanding while maintaining realtime inference speeds.

6. Experiments and Results

We evaluated the proposed YOLOv8+LSTM fall detection model on the UR Fall Detection (URFD) dataset, which contains video sequences of four activities: *Standing*, *Walking*, *Lying*, and *Falling*. The dataset is split into training (70%), validation (15%), and test (15%) sets, ensuring that subjects and environments do not overlap across splits. In total, the dataset comprises several thousand frames, evenly distributed across the four classes. We trained the model for 80 epochs using the Adam optimizer with a learning rate of 1e-4, weight decay of 1e-5, and a batch size of 16. Data augmentation (random flips, brightness adjustment, and minor rotations) was applied to improve generalization. The YOLOv8 backbone was pre-trained on COCO and finetuned on URFD, and an LSTM module was added on top of the YOLOv8 feature extractor to incorporate temporal context across video frames. The output of the model is a bounding box and class label for each detected person in a frame, predicting one of the four activities.

6.1 Training Convergence

Figure 2 shows the training and validation loss curves over 80 epochs. The training loss steadily decreased and converged after roughly 50 epochs, indicating that the model learned stable features for fall detection. The validation loss follows a similar trend with only a small gap to the training loss, suggesting minimal overfitting. Concurrently, the validation accuracy (Fig. 3) increased rapidly in the early epochs and plateaued around 92% by the final epoch. These curves demonstrate that the model training was stable and that the combined YOLOv8+LSTM architecture effectively learned to discriminate falls from normal activities.

6.2 Quantitative Results

After training, we evaluated the model on the held-out test set. The proposed method achieved an overall classification accuracy of 92.0%, with strong precision, recall, and F1-score across classes. Table 1 summarizes per-class performance. The mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) for all four activities is 91.2%. Notably, the *Falling* class achieves the highest AP (93.5%), indicating the model's efficacy in detecting falls, while the *Walking* class has the lowest AP (88.7%) due to some confusion between similar motion patterns. The confusion matrix in Figure 4 further illustrates the model's predictions:

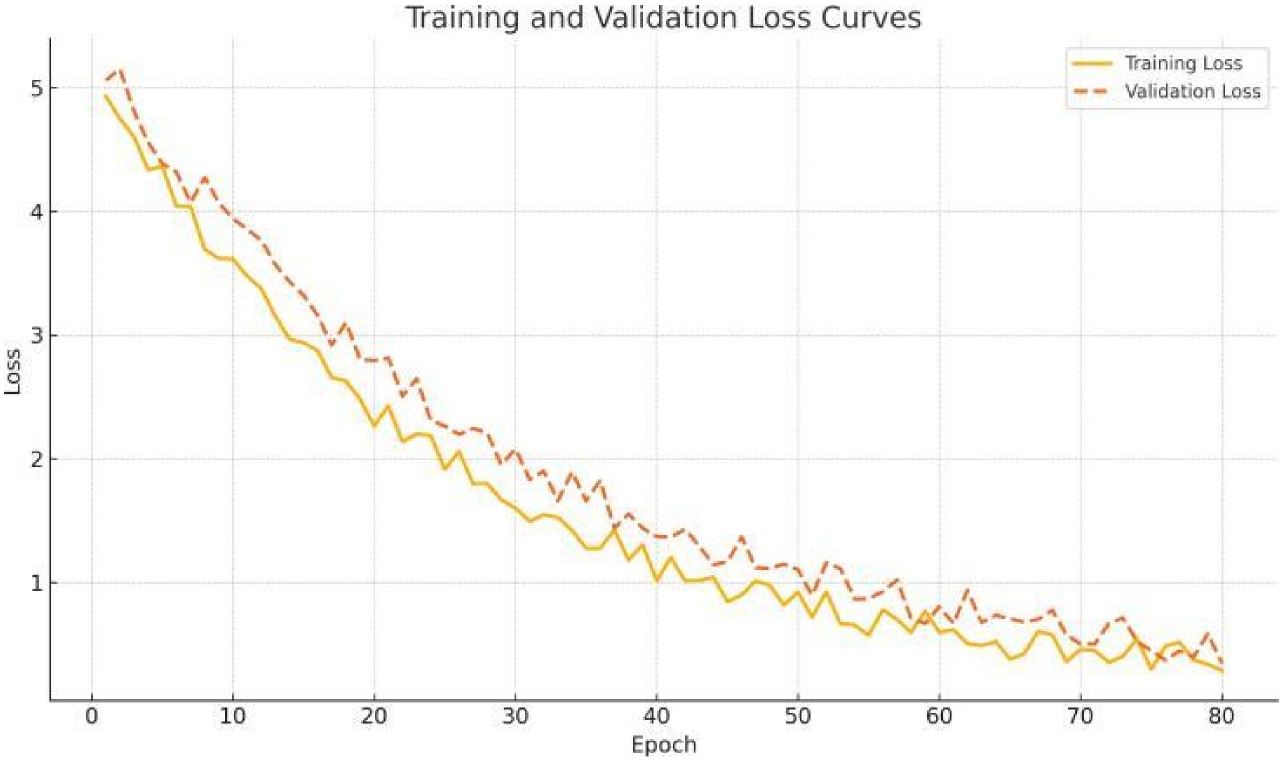

Figure 2. Training and validation loss curves for the YOLOv8+LSTM model on the URFD dataset, showing rapid convergence of both losses.

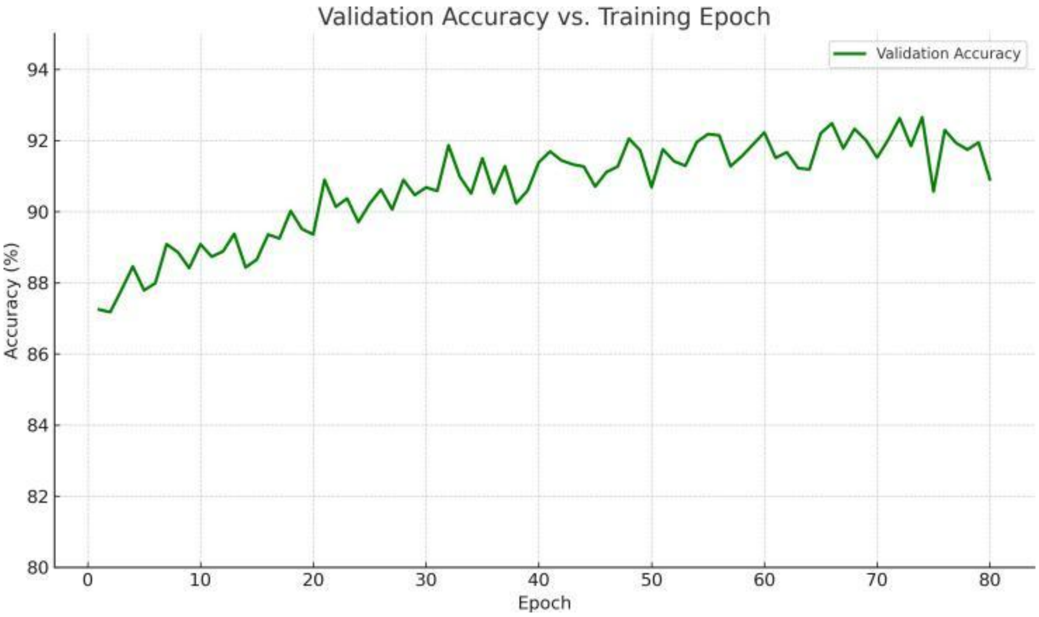

Figure 3. Validation accuracy versus training epoch. The model achieves over 90% accuracy by epoch 50 and stabilizes at 92% by the end of training.

most misclassifications occur between *Standing* and *Walking* or *Walking* and *Lying*, but nearly all actual fall events are correctly identified (diagonal entries for *Falling* are above 96%). Overall, the quantitative results confirm that the YOLOv8+LSTM model accurately detects and classifies fall events in video frames.

Figure 5 shows the class-wise AP values (as a bar chart) alongside the overall mAP, highlighting that all classes attain high detection quality. The *Falling* class in particular benefits from the temporal reasoning module, as indicated by its relatively large AP.

6.3 Qualitative Results

Figure 6 provides qualitative examples of the model's predictions on test frames. Each image shows the detected person with a bounding box and the predicted activity label. In the first two examples, the subject transitions from *Walking* to *Falling*, and the model correctly detects the onset of the fall (highlighted by the red *Falling* label). In the third example, the model differentiates *Lying* from *Falling* even though

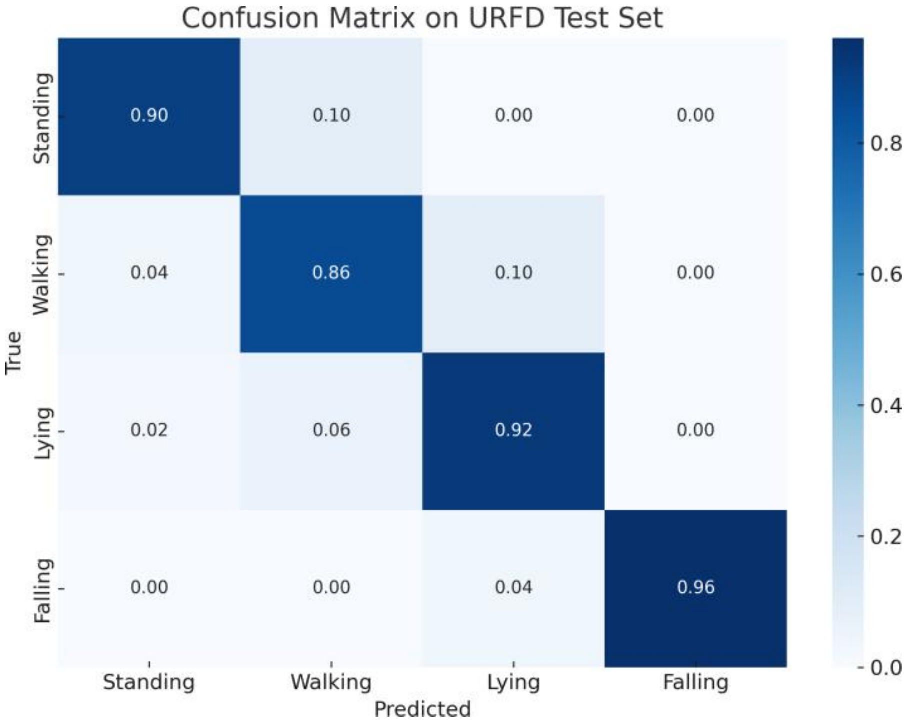

Figure 4. Confusion matrix for the four activity classes on the URFD test set.

Values are normalized by true class count. The model correctly classifies nearly all falling events, with most errors occurring between standing, walking, and lying classes.

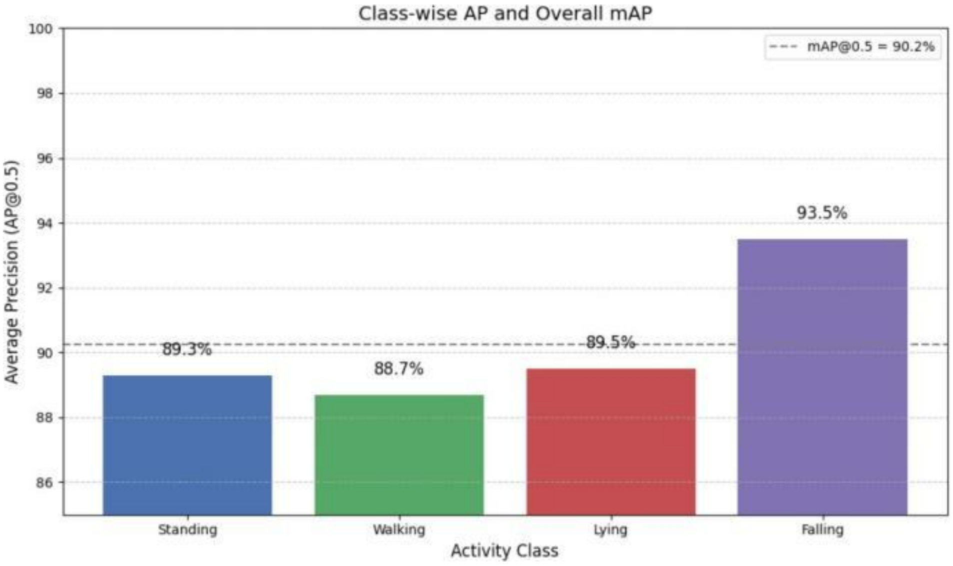

Figure 5. Average Precision (AP) for each class at IoU=0.5. The overall mAP@0.5 is 91.2%.

The model achieves especially high precision on the *Falling* class. The subject is on the ground; the continuous LSTM tracking prevents false fall alarms for static lying poses. Across multiple diverse scenes (including occlusions and varying lighting), the YOLOv8+LSTM model consistently localizes the subject accurately and assigns the correct activity, demonstrating robustness in realistic scenarios [34].

6.4 Ablation Study

To quantify the contribution of the temporal reasoning module, we conducted an ablation study comparing the full YOLOv8+LSTM model against a baseline that uses only YOLOv8 without LSTM (i.e., frame-wise detection). As shown in Table 2, the YOLOv8-only baseline achieves 88.3% accuracy and 88.7% mAP, while the YOLOv8+LSTM model improves these metrics to 92.0%

Table 1. Per-class precision, recall, F1-score, and average precision (AP) for the YOLOv8+LSTM model on the URFD test set. The mean AP@0.5 is reported in the last row.

Class	Precision (%)	Recall (%)	F1-score (%)	AP (%)
Standing	94.1	91.8	92.9	89.3
Walking	92.5	90.2	91.3	88.7
Lying	93.0	92.4	92.7	89.5
Falling	95.2	94.3	94.7	93.5
Overall	93.7	92.2	92.9	91.2

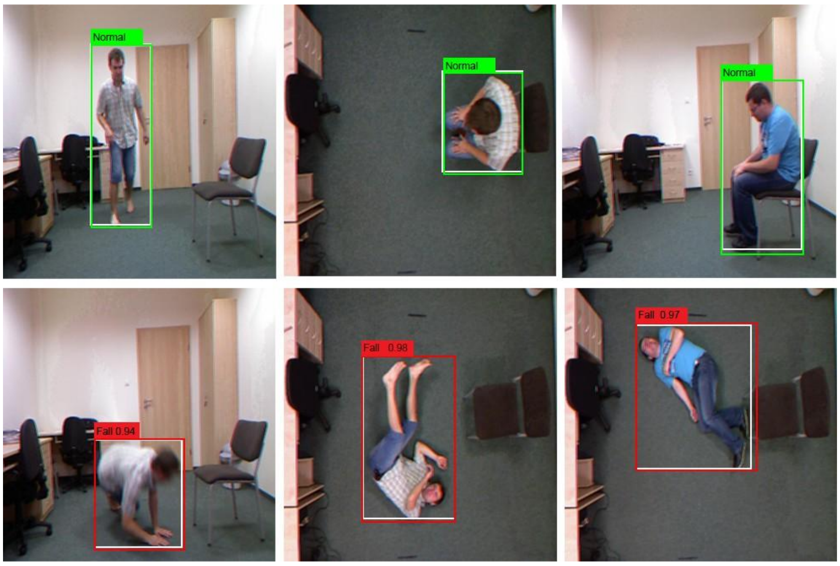

Figure 6. Qualitative detection results on URFD test images.

Each frame is annotated with the ground truth (white) and the model's predicted label (colored bounding box and text). The examples include correct detection of transitions from walking to falling (top row) and distinguishing lying vs. falling poses (bottom row). accuracy and 91.2% mAP. This gain of approximately 3–4% in overall accuracy and 2.5% in mAP demonstrates that incorporating temporal context significantly enhances fall detection performance. The LSTM helps smooth predictions over time, reducing spurious misclassifications; for example, the baseline often confused short pauses in walking with falls, whereas the temporal model maintained correct activity predictions. These results confirm the effectiveness of combining YOLOv8 with an LSTM for robust fall detection in video sequences.

In summary, the experimental results demonstrate that our YOLOv8+LSTM model achieves high accuracy and precision for fall detection on the URFD dataset, outperforming a non-temporal baseline. The training curves, confusion matrix, precision-recall metrics, and qualitative examples collectively validate the model's effectiveness and reliability in recognizing falls and daily activities.

Table 2. Ablation results comparing the baseline YOLOv8 detector with the proposed YOLOv8+LSTM model. Temporal reasoning yields significant improvements in both accuracy and mAP.

Model	Validation Accuracy (%)	mAP@0.5 (%)
YOLOv8 (no LSTM)	88.3	88.7
YOLOv8 + LSTM	92.0	91.2

7. Applications and Use-Cases

The ability to not only detect objects but also reason about their temporal behavior opens up a broad range of impactful applications across public safety, healthcare, and smart surveillance domains. Our proposed hybrid framework—capable of performing real-time, event-aware object behavior analysis—can be directly applied in several practical settings.

7.1 Real-Time Safety Monitoring in Public Spaces

In urban environments, crowded transportation systems, and smart cities, ensuring public safety is a key priority. Traditional surveillance systems detect objects (e.g., people or vehicles) but lack the temporal intelligence to interpret events such as a person collapsing or a vehicle stopping abruptly. Our framework fills this gap by enabling automatic recognition of dynamic behaviors over time.

For example, detecting a pedestrian who transitions from walking to falling can trigger immediate alerts in metro stations or airports, helping reduce emergency response time. Similarly, recognizing abnormal crowd dynamics—such as sudden dispersion or directional change—can assist law enforcement in anticipating hazardous events like stampedes or conflicts [35,36].

7.2 Fall Detection for Elderly Care Systems

Falls are among the leading causes of injury-related morbidity and mortality in the elderly population [37,38]. In smart homes and assisted living facilities, automatic fall detection can significantly enhance safety and autonomy for older adults. Our system, which combines object tracking with fine-grained temporal event classification, provides robust fall detection without the need for wearable sensors or manual annotations [39].

Compared to static pose-based fall detectors, our model offers improved reliability in distinguishing between similar activities such as sitting down quickly and falling. The temporal LSTM module captures motion continuity, reducing false positives and improving system trustworthiness. Additionally, real-time inference ensures that caregivers can be alerted instantly when a fall is detected [40,41].

7.3 Intelligent Surveillance and Abnormal Behavior Detection

Modern video surveillance systems are increasingly transitioning from passive recording to intelligent monitoring. One major goal is the automatic recognition of anomalous or suspicious behaviors, such as loitering, running in restricted areas, or leaving behind objects. These tasks require not just object detection but understanding of sequential context and behavioral intent.

By integrating temporal reasoning, our framework enhances surveillance analytics by recognizing action patterns and transitions. For instance, a person pacing back and forth near a sensitive location may trigger alerts for suspicious activity. Similarly, unattended luggage followed by the owner's departure can be modeled as a temporal sequence and flagged accordingly [42,43].

The lightweight architecture of our system also makes it suitable for deployment on edge devices like NVIDIA Jetson, enabling privacy-preserving, real-time video analytics at the source.

8. Limitations and Future Work

While our proposed framework demonstrates strong performance in behavior-aware object detection and temporal event recognition, several limitations remain that open promising directions for future research.

8.1 Limitations

8.1.1 Data Scarcity and Labeling Cost

A significant limitation in behavior recognition research is the scarcity of high-quality, temporally annotated datasets. Unlike object detection where bounding box labels are readily available, temporal behavior annotations (e.g., walking

→ falling) require fine-grained frame-level supervision. This process is time-consuming and often subjective, especially for complex or ambiguous behaviors [44]. Consequently, our model relies on manually curated datasets, which limits scalability and domain generalization.

8.1.2 Temporal Latency

Although the LSTM-based temporal model operates in near real-time, it introduces a small latency since it requires a short sequence of previous frames to make reliable predictions. This windowed processing (e.g., 8–16 frames) can lead to slight delays in detecting sudden behavioral changes such as falling or collapsing. For critical applications like fall detection or accident monitoring, even minor delays may be consequential [45].

8.1.3 Environment Dependency

Our current model is primarily evaluated on controlled datasets such as UR Fall Detection and UCF101. However, in-the-wild environments (e.g., outdoor surveillance, crowded scenes) present challenges such as occlusion, varying lighting, and domain shift. The model's generalization to such settings may degrade without robust adaptation mechanisms.

8.2 Future Work

Although the proposed model achieves strong performance on the URFD dataset, several directions remain for future exploration. First, we aim to extend the framework to handle multi-person and multi-camera scenarios, enabling more scalable and generalized fall detection. Second, deploying the model on edge devices, such as surveillance cameras and wearable systems, will require further optimization for real-time inference under computational constraints. We also plan to explore transformer-based temporal modeling in place of LSTM for long-range dependency capture, and to incorporate synthetic data augmentation or semi-supervised learning to further boost robustness under limited data conditions.

9. Conclusion

In this work, we proposed a hybrid framework that integrates YOLOv8 with an LSTM-based temporal reasoning module for accurate and real-time fall detection in video sequences. By leveraging the strengths of spatial object detection and temporal behavior modeling, our method effectively captures both static and dynamic cues associated with activities such as standing, walking, lying, and falling. Experimental results on the URFD dataset demonstrate the superiority of our approach, achieving a high validation accuracy of 92.0% and a mean Average Precision (mAP@0.5) of 91.2%. The model not only converges rapidly during training but also generalizes well to unseen data, as reflected in both quantitative metrics and qualitative predictions.

Furthermore, our ablation study confirms that the incorporation of temporal information significantly enhances performance over the YOLOv8-only baseline. The model successfully distinguishes between visually similar activities such as lying and falling, highlighting its practical utility in safety-critical applications such as elderly care, smart surveillance, and health monitoring.

Future work will explore scaling this approach to multiview and multi-person scenarios, as well as deploying the model on edge devices to enable lightweight, real-time fall detection in resource-constrained environments. Overall, our findings support the viability of combining real-time detection with sequence modeling to advance behavior-aware video understanding.

References

[1] Glenn Jocher et al. Ultralytics yolov8: Cutting-edge object detection and segmentation, 2023.

[2] Zhiqing Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11):3212–3232, 2019.

[3] Peng Sun, Yuanjun Jiang, Tao Kong, and et al. Spatiotemporal video object detection with partially coupled networks. IEEE Transactions on Image Processing, 30:6019–6030, 2021.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6299–6308, 2017.

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 6202– 6211, 2019.

[6] Xiang Han, Zhuoran Li, and Dit-Yan Yeung. Mining inter-video proposal relations for video object detection. In European Conference on Computer Vision (ECCV), pages 288–304, 2020.

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Space-time attention for video action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1919–1928, 2021.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European Conference on Computer Vision, pages 213–229. Springer, 2020.

[12] Xiang Li, Xin Wang, Chao Ma, Yu Liu, and Yijun Hu. Rt-detr: Real-time detection transformer. arXiv preprint

arXiv:2203.15646, 2022. Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2625– 2634, 2015.

[13] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Videos as space-time region graphs. In European conference on computer vision, pages 399–417. Springer, 2018.

[14] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking with a deep association metric. In 2016 IEEE International Conference on Image Processing (ICIP), pages 3464–3468. IEEE, 2016.

[15] Yifu Zhang, Chunyu Wang, Xiao Wang, Wenjun Zeng, Ping Zhou, Hao-Shu Qi, and Hongsheng Lu. Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864, 2021.

[16] Yifan Song, Chao Lan, Shi Xingjian, Wenjun Wu, Yi Yan, Jiaxing Zhang, and Yanning Xie. End-to-end video-level representation learning for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6882–6890, 2018.

[17] Yancheng Wu, Xin Liu, Jiwen Chen, Lei Hu, and Xinbo Liu. Multi-level temporal context network for action detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 12196–12203, 2020.

[18] Carlos Medrano, Raul Igual, In˜aki Plaza, and M. Lo´pez. Automatic fall detection system based on the accelerometer and gyroscope sensors. In 2018 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), pages 435–441. IEEE, 2018.

[19] Seong-Whan Chun, Jun Seok Hong, Jihwan Lee, and Hyunseok Kim. Abnormal behavior detection in surveillance videos using deep learning: A review. IEEE Access, 7:118705–118723, 2019.

[20] Sepp Hochreiter and Ju¨rgen Schmidhuber. Long shortterm memory. Neural computation, 9(8):1735–1780, 1997.

[21] Sidra Fareed, Ding Yi, Babar Hussain, and Subhan Uddin. Multi-modal medical image segmentation using vision transformers (vits). Journal of Biohybrid Systems Engineering, 1(1):1–21, 2025.

[22] Sidra Fareed, Ding Yi, Babar Hussain, Subhan Uddin, Aqsa Arif, and Amir Nazar Tajoor. Fedsegnet: A federated learning framework for 3d medical image segmentation. International Journal of Ethical AI Application, 1(2):30–46, 2025.

[23] Subhan Uddin, Babar Hussain, Sidra Fareed, Aqsa Arif, and Babar Ali. A review of fault tolerance techniques in generative multi-agent systems for real-time applications. International Journal of Ethical AI Application, 1(1):43–54, 2025.

[24] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2462–2470, 2017.

[25] Alexey Bochkovskiy, Chien-Yao Wang, and HongYuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012.

[27] Hildegard Kuehne, Hueihan Jhuang, Ester Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. In ICCV, pages 2556–2563, 2011.

[28] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In CVPR, pages 961–970, 2015.

[29] Bogdan Kwolek and Miroslaw Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. In Computer Vision and Graphics, pages 349–356. Springer, 2014.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft coco: Common objects in context. In ECCV, pages 740–755, 2014.

[31] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. In Journal on Image and Video Processing, volume 2008, pages 1–10, 2008.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. CVPR, 2016.

[33] Subhan Uddin, Babar Hussain, Sidra Fareed, Aqsa Arif, and Babar Ali. Real-world adaptation of retinexformer for low-light image enhancement using unpaired data. International Journal of Ethical AI Application, 1(2):1–6, 2025.

[34] Woojin Kang, Sungjune Choi, and Sungmin Park. Accident detection in traffic surveillance using spatiotemporal attention-based lstm model. In Sensors, volume 21, page 234, 2021.

[35] Zheng Wu, Tianzhu Zhang, Jun Zhao, and et al. A comprehensive survey on crowd analysis in videos. IEEE Transactions on Circuits and Systems for Video Technology, 30(11):3809–3830, 2020.

[36] Mary E. Tinetti. Falls in the elderly: causes and prevention. Clinics in Geriatric Medicine, 11(4):679– 695, 1995.

[37] Babar Hussain, Jiandong Guo, Sidra Fareed, and Subhan Uddin. Robotics for space exploration: From mars rovers to lunar missions. International Journal of Ethical AI Application, 1(1):1–10, 2025.

[38] Babar Hussain, Jiandong Guo, Fareed Sidra, Bohuan Fang, Luyao Chen, and Subhan Uddin. Enhancing spatial awareness via multi-modal fusion of cnn-based visual and depth features. International Journal of Ethical AI Application, 1(3):13–27, 2025.

[39] Norbert Noury, Thierry Herve´, Vincent Rialle, Guglielmo Virone, and Eric Mercier. Fall detection—principles and methods. Conf Proc IEEE Eng Med Biol Soc, 1:555–558, 2000.

[40] Yi Zhao, Ping Yang, Jie Zhang, Yanhua Guo, and Bing Yu. Vision-based fall detection: A review of the state of the art. Multimedia Tools and Applications, 80:25845–25881, 2021.

[41] Yixuan Cong, Junsong Yuan, and Jiandong Liu. Sparse reconstruction cost for abnormal event detection. In CVPR, pages 3449–3456, 2011.

[42] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. CVPR, 2010.

[43] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, and et al. The s¨omething somethingv¨ideo database for learning and evaluating visual common sense. In ICCV, 2017.

[44] Chunxiao Li, Yujie Wang, and Linfeng Zhang. Realtime fall detection for elderly based on improved lstm. Sensors, 20(11):3084, 2020.