Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00381-8

Original Article

A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments

Rabbi Mushad [1], Md Akhtaruzzaman [2, 3]

1. *Computer Science and Engineering, Military Institute of Science and Technology, Dhaka, BGD*
2. *Robotics and Mechatronics Engineering, Daffodil International University, Dhaka, BGD*
3. *Research and Development, Quantum Robotics and Automation Research Group (QRARG), Dhaka, BGD*

Received: September 01, 2026 | Review began: September 04, 2026 | Review ended: November 18, 2026 | Published: November 19, 2026

© **Copyright** 2026
This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Surveillance in controlled environments such as military bases and other restricted facilities requires timely detection of suspicious activities. This paper presents a real-time multi-mode deep learning framework that integrates YOLOv8-based person detection, Deep SORT multi-object tracking, spatio-temporal feature extraction, and behavior recognition through rule-based, supervised, and self-learning operating modes. A six-mode ablation study compares hybrid and no-fallback variants under a shared deployment pipeline. Tracking evaluation on the PETS09 surveillance dataset confirms stable person trajectories suitable for downstream analysis, and a two-stage sensitivity study identifies operational thresholds for loitering duration, radius, and running speed. Under independent manual ground-truth validation on a controlled deployment video, the self-learning hybrid mode achieved the highest accuracy and macro F1 among the six evaluated modes, while supervised-synthetic and rule-based modes remain competitive. Removing the rule-based fallback causes both LSTM-only ablations to degrade toward near-random behavior, demonstrating that fallback fusion is functionally essential. Self-learning achieves this performance without any manually labeled deployment data, suggesting potential practical applicability for controlled surveillance environments, pending further validation. Preliminary cross-scene evaluation on a second deployment video shows that all hybrid modes maintain high agreement with a rule-based reference and reproduce the same no-fallback degradation pattern across scenes. On the manually annotated deployment subset, the self-learning hybrid mode attains an accuracy of 0.899 and a macro F1 of 0.713, while both no-fallback ablations collapse to near-random accuracy of approximately 0.50, a 38-40 percentage-point accuracy drop across the two ablations relative to their fallback-enabled counterparts, indicating that rule-based fallback fusion is necessary for maintaining above-chance performance in this framework.

Categories: AI applications, Computer Vision

Keywords: real-time video surveillance, suspicious behavior detection, deep learning, multi-object tracking, controlled surveillance environments

Introduction

Automated video surveillance is increasingly deployed in security-critical environments [1,2], where suspicious behavior detection is a core requirement [3-5]. In controlled environments such as military facilities, secure perimeters, and other restricted areas, missed alarms directly reduce operational effectiveness and increase security risk [6,7]. Traditional surveillance depends heavily on human operators and suffers from fatigue, distraction, and limited sustained attention over long periods [1,2].

Most existing behavior recognition systems are designed for open public spaces, where motions such as running or loitering are only weakly indicative of threats [2,8]. In controlled environments, the same actions can have severe implications: loitering near an access point may signal reconnaissance, while sudden running may indicate an escape attempt or protocol violation [8,9]. This contextual difference motivates specialized surveillance solutions designed for controlled environments [6,10].

Methodologically, prior works often rely on hand-tuned rule-based systems that are interpretable but fragile across deployments [9,10]. Supervised temporal models capture richer kinematic patterns but require manual annotations that are rarely available in controlled surveillance environments [11-13]. Recent annotation-free approaches in smart city surveillance mitigate labeling costs [4,14], yet they remain largely untested in controlled surveillance contexts with narrower, operationally defined suspicious behaviors.

To address these gaps, this paper proposes a modular multi-mode deep learning framework for suspicious behavior detection in controlled environments. Rather than introducing a new detector or temporal architecture, the framework unifies three complementary operating modes on a shared detection-tracking-feature backbone: an interpretable kinematic rule-based mode for zero-label operation, a supervised Long Short-Term Memory (LSTM)-based mode for labeled scenarios, and an annotation-free self-learning mode that adapts using rule-derived pseudo-labels [4,9,13]. This design allows dynamic mode selection and hybrid fusion based on data availability and operational constraints.

The framework is validated using two complementary evaluation protocols: accuracy against human-annotated ground truth on a manually labeled subset and agreement with deterministic rule-based logic over full deployment sequences. The main research question is how to design and evaluate a multi-mode framework that reliably detects motion-based suspicious activities in controlled surveillance environments without manual deployment labels. The specific objectives are to (i) integrate person detection, tracking, feature extraction, and behavior classification into a modular pipeline; (ii) instantiate rule-based, supervised, and self-learning operating modes on a common backbone; and (iii) quantify the impact of mode choice and rule-based fallback fusion on behavior recognition performance. The framework is evaluated on surveillance footage from controlled indoor and outdoor night environments using person-centric tracks as the primary units of analysis.

Related work

Video anomaly detection and suspicious behavior recognition are key components of intelligent surveillance systems, as manual monitoring does not scale to large, real-time deployments [15,16]. Recent work spans supervised, weakly supervised, unsupervised, and adaptive learning paradigms across diverse surveillance settings, but common challenges remain, including limited annotated data, strong scene dependence, and difficulty in defining suspicious behavior consistently across real-world environments [3,16].

Early studies investigated supervised suspicious activity analysis in controlled or semi-controlled academic, campus, and examination environments, showing that temporal modeling can distinguish suspicious from normal behavior in constrained scenes [17,18]. Other research extended suspicious behavior recognition beyond body motion by incorporating facial and physiological indicators [10]. These studies demonstrate the value of context-specific analysis, but they typically use task-specific pipelines instead of flexible frameworks that can adapt under different annotation conditions.

As deep visual feature learning matured, many approaches adopted integrated detection-tracking-temporal pipelines. Hybrid architectures such as YOLO-SORT-LSTM, time-distributed Convolutional Neural Networks (CNNs), and Conv3D-based anomaly detectors combine object localization with temporal modeling to improve behavior recognition in dynamic scenes [12,19]. These methods still struggle with visually similar actions, sensitivity to scene changes, and brittle performance when training and deployment distributions do not match [7,13].

A strong recent trend is the use of YOLO-based detector-centered pipelines because they offer a practical balance between accuracy and speed. Studies have combined YOLO variants with tracking, dense crowd analysis, multi-view fusion, and infrastructure monitoring for real-time abnormal or suspicious event detection [20-23]. While these systems

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

demonstrate effective real-time deployment, their performance often remains sensitive to occlusion, narrow training distributions, and narrowly defined event taxonomies [20,21].

Studies on loitering detection, crowd monitoring, and anomaly recognition highlight that effective behavior analysis depends on matching the model design to the operational context, especially when behaviors such as loitering or irregular motion have domain-specific meanings [24,25]. This is particularly relevant in controlled surveillance environments, where suspicious behaviors are narrowly defined and closely linked to security procedures rather than broad public anomalies.

More recent efforts move beyond closed-set anomaly detection by supporting richer anomaly categories and considering deployment aspects such as adaptability and system design [14,16,26]. Overall, the literature has progressed from early CNN and CNN-Recurrent Neural Network systems to integrated detection-tracking-temporal pipelines and efficient YOLO-based designs [21,26]. Several gaps remain for controlled surveillance: limited flexibility across annotation conditions (zero-label, supervised, and self-learning), sensitivity to scene shift, limited support for operationally defined suspicious behaviors, and a lack of evaluation protocols that separate deployment-reference agreement from independent ground-truth validation [16,25]. The present work addresses these gaps by proposing a detector-centered multi-mode framework that unifies rule-based, supervised, and self-learning modes and explicitly evaluates both fallback fusion and cross-scene transfer in controlled environments.

Clarification on "Multi-Mode" Terminology

It is clarified here that "multi-mode" in this work refers to three interchangeable operating modes (rule-based, supervised LSTM, and self-learning) that share a single motion-based detection-tracking-feature pipeline, not a "multimodal" architecture that fuses heterogeneous sensor inputs (e.g., RGB, audio, skeleton, or depth streams). The proposed contribution is therefore best positioned against single-modality, motion/trajectory-based detector-tracker-temporal pipelines such as YOLO-SORT-LSTM and related hybrid architectures, rather than multimodal fusion frameworks, which address a different problem (combining heterogeneous input streams) from the one targeted here (flexible operation under differing label-availability conditions on a single motion-based input stream). Relative to this more directly comparable class of methods, the contribution of this work is not a new detector or temporal network, but the explicit unification of zero-label, supervised, and self-learning classification strategies on a shared backbone, together with a controlled ablation isolating the necessity of rule-based fallback fusion. This is a design and evaluation question that, to the authors' knowledge, has not been directly addressed in the reviewed literature on controlled-environment surveillance.

Materials And Methods

System architecture overview

The proposed framework follows a modular pipeline (Figure *1*).

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

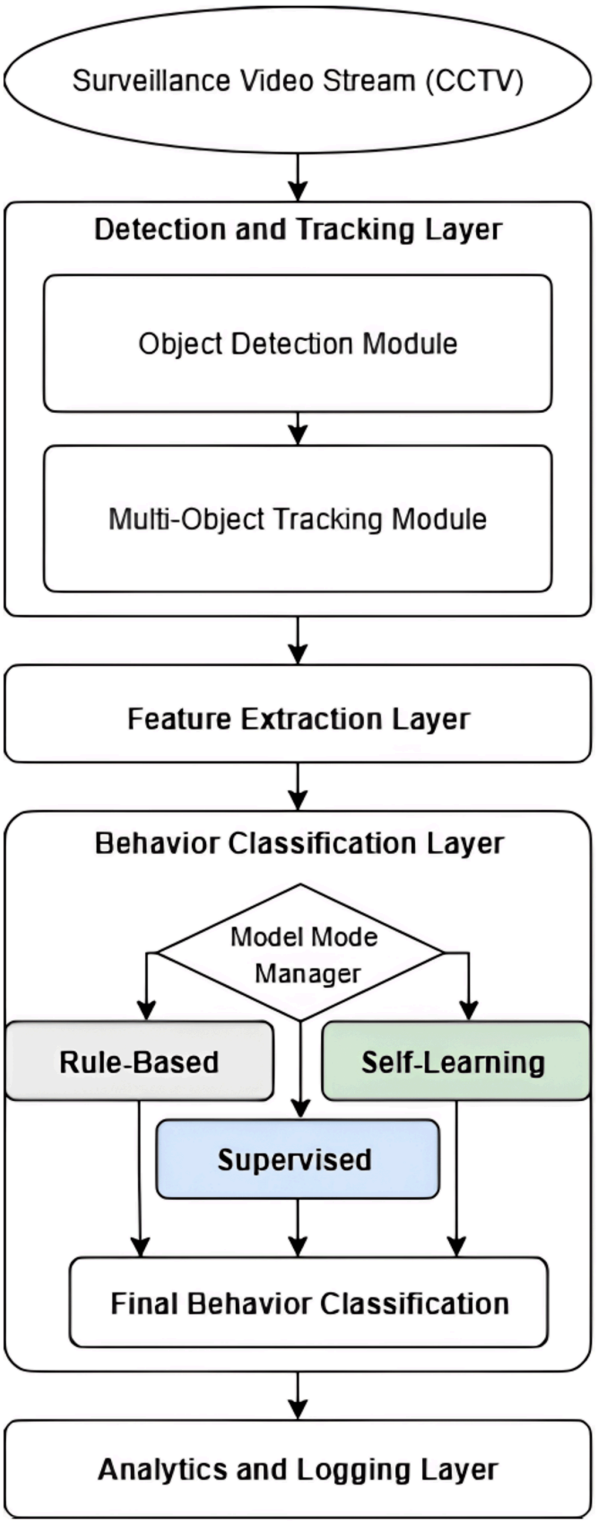


FIGURE 1: System Architecture Diagram

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

It processes raw video through four sequential stages: person detection, multi-object tracking, spatio-temporal feature extraction, and behavior classification. All operating modes share the same detection, tracking, and feature extraction backbone. Only the final classification step changes across modes. This design separates the perception backbone from the decision logic. It allows an operator to switch between an interpretable rule-based mode when auditability is required, a supervised mode when labeled data are available, or a self-learning mode when deployment data are unannotated, without changing any hardware or low-level processing component. The modular structure also supports controlled ablation: no-fallback variants remove only the rule-based decision path, so observed performance differences reflect classification strategy rather than changes in perception quality. The framework is implemented using standard deep learning and computer vision libraries.

Detection and tracking

Person detection uses YOLOv8-small pretrained on the COCO dataset, retaining only the person class with a confidence threshold of 0.5 to suppress weak detections. Multi-object tracking uses Deep SORT, which applies Kalman filtering for motion prediction and appearance-based matching for association, assigning each person a stable track identity across frames.

Feature extraction

For each confirmed track, spatio-temporal descriptors are computed over a fixed sequence of N = 20 frames. The derivation of the feasible range for N is provided in the Appendices (Derivation of the Sequence Length N section). The seven-dimensional feature vector includes normalized centroid coordinates (x, y), instantaneous speed, acceleration, movement radius, temporal index, and a motion consistency score, following the trajectory representation of Akhtaruzzaman et al. [27]. This compact design captures both short-term motion and longer-term tendency without requiring camera calibration.

Behavior modeling

Two complementary behavior modeling components are implemented: a rule-based module and a learning-based LSTM module.

Rule-Based Module: Tracks are described using centroid positions and per-frame displacement, measured in pixels per frame at 25 frames per second (FPS). Running is flagged when sustained displacement exceeds a velocity threshold; loitering is flagged when long-term presence is detected within a small radius around a reference position. The rule set applies simple inequalities over displacement from a reference frame, yielding one of three labels: Normal, Loitering, or Running. These same rules are reused as pseudo-label sources for self-learning.

LSTM Module: The learning-based branch uses an attention-enhanced LSTM that maps the 20 × 7 feature sequence to three class scores. The network has two stacked LSTM layers (hidden size 32), a multi-head attention block with a residual connection and layer normalization, and three fully connected layers (32→32→16→3) that output raw class logits over Normal, Loitering, and Running; softmax normalization is applied during inference and internally by the cross-entropy loss during training. The model is trained in PyTorch using Adam (learning rate = 1.5e-4), cross-entropy loss with label smoothing, gradient clipping, dropout, and early stopping on validation loss.

Multi-mode operation

A mode manager coordinates three operating settings on the shared backbone (Table *1*).

Mode	Description
Rule Based	Uses only kinematic thresholds; produces interpretable decisions
Supervised	Uses the LSTM classifier trained on labeled trajectories

Self-Learning	Periodically updates the LSTM from rule-derived pseudo-labels without manual annotation

TABLE 1: Operating Modes

Operating modes coordinated by the mode manager on the shared detection–tracking–feature backbone.
LSTM, Long Short-Term Memory

In learning-based modes, a fusion rule compares the maximum LSTM class probability to a fixed confidence threshold (0.5). When this probability meets or exceeds the threshold, the LSTM output is used; otherwise, the rule-based decision is selected as a fallback. For the Running class specifically, an additional velocity check is applied: an LSTM prediction of Running is only accepted if the track's instantaneous speed at that frame exceeds 140 px/s; below this speed, the rule-based decision is used regardless of LSTM confidence, thereby reducing false Running classifications near the decision boundary.

The self-learning retraining loop, in which pseudo-labeled samples accumulate in a buffer and are used to fine-tune the LSTM once a minimum sample count is reached, proceeds as follows:

1. Initialize the LSTM model M with pre-trained weights.
2. Set the fusion threshold τ = 0.5 and the buffer threshold B_{thresh} = 300.
3. Initialize an empty buffer B.
4. For each frame t: (a) detect and track persons (YOLOv8 + Deep SORT); (b) extract the features $f_i(t)$ and the rule label $y_{\text{Rule}}(t)$; (c) append $(f_i(t), y_{\text{Rule}}(t))$ to B; (d) if $|B| \geq B_{\text{thresh}}$, balance the classes and add Gaussian noise, fine-tune M on B for E epochs, and reset B to an empty set.

The retraining loop operates as follows: during inference, every track prediction produced by the rule-based module is logged alongside its corresponding 20 × 7 feature sequence. Before being accepted into the buffer, each candidate sample must satisfy class-specific consistency checks over its recent trajectory window (e.g., bounded speed and movement-radius variance for Normal and Loitering candidates, and a minimum count of consecutive high-speed frames for Running candidates), which filter out sequences with unstable or ambiguous kinematic signatures prior to pseudo-label acceptance. These rule-derived pseudo-labels accumulate in a rolling buffer; once the buffer reaches the minimum threshold of 300 samples, a fine-tuning pass is triggered on the current LSTM weights using 18 epochs (buffer below 500 samples) or 22 epochs (otherwise) with early stopping (patience = 6) to avoid overfitting to a single retraining batch. Retraining is repeated every 1,200 frames to allow the model to adapt to evolving trajectory patterns without retraining on every incoming sample, which would be computationally prohibitive on CPU-only hardware. Retraining additionally requires that all three behavior classes (Normal, Loitering, Running) be represented by at least 30 samples in the current buffer, and is deferred until this class-balance condition is met if a scheduled checkpoint arrives early. When retraining is triggered, class counts in the buffer are balanced by oversampling underrepresented classes up to the size of the largest class; oversampled copies receive additive Gaussian noise (sigma = 0.015, clipped to [−3.0, 3.0]) as a light augmentation to reduce the risk of the LSTM memorizing duplicated samples rather than learning genuine motion variation. Because the rule-based module determines the pseudo-labels, the self-learning mode is best understood as distilling the rule-based decision boundary into a learned representation that additionally benefits from temporal aggregation across the 20-frame window, rather than discovering new behavioral categories independently.

Analytics and logging

A logging module records frame index, track identifier, bounding box, velocity, predicted label, and confidence score for each track. Analysis scripts generate plots, confusion matrices, and event summaries for operator review and evaluation.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Computing environment

All experiments were run on a laptop with an Intel Core i5-8265U CPU, 8 GB RAM, and integrated Intel UHD graphics (no discrete GPU), using Python 3.10, PyTorch 2.0, and OpenCV 4.8 on Windows 11 Pro. On this CPU-only setup, the end-to-end pipeline operated below real-time frame rates; full 25-30 FPS operation is therefore expected only under GPU deployment.

Datasets

Full dataset construction details, including the annotation protocol (Label Studio), inter-annotator agreement (Appendices Table *12*), class distribution across Normal/Loitering/Running (Appendices Table *11*), and the derivation of the feature sequence length N (Appendices, Derivation of the Sequence Length N section), are documented in the Appendices for reproducibility. The rationale for component selection (YOLOv8-small for detection, Deep SORT for tracking, and the seven-dimensional trajectory feature vector) follows established practice in detector-tracker-temporal pipelines, balancing detection/tracking reliability against the computational constraints of CPU-only deployment (Computing Environment section).

The framework uses complementary datasets for detector verification, tracking evaluation, supervised behavior modeling, and deployment-time adaptation. Penn-Fudan [28] was used to confirm that the pretrained YOLOv8-small detector provides reliable pedestrian localization for downstream tracking. PETS09-S2L1 [29] served as a controlled multi-object tracking benchmark with an elevated static camera and low-pedestrian-density surveillance conditions.

For behavior modeling, CUHK Avenue [30] provided a public benchmark containing normal and abnormal events, including running and loitering. In addition, a synthetic dataset of 4,700 Gaussian-sampled sequences (2,000 Normal, 1,500 Loitering, 1,200 Running) was generated to pre-train the LSTM under class-balanced, rule-consistent motion patterns. As the simulator controls both train and test splits, its accuracy is treated only as a convergence check and is not reported as a main result.

Unlike the publicly available benchmark datasets used for detection and tracking validation (Penn-Fudan, PETS09-S2L1, CUHK Avenue), the primary and cross-scene deployment videos were recorded specifically for this study and are not public data; recording was conducted with written informed consent, as detailed in the Ethics Statement and Conflict of Interest Disclosures section.

Deployment adaptation and primary evaluation used a 92-s indoor deployment video (25 FPS, 1,920 × 1,080, 2,294 frames) recorded from a static elevated camera with three operational behaviors (Normal, Loitering, Running). A manually annotated subset of 4,392 frame-person instances was created, of which 4,330 samples (2,183 Normal, 1,987 Loitering, 160 Running) were matched to confirmed Deep SORT tracks and used in all manual-ground-truth evaluation. Matching required that each annotated frame-person instance (identified by frame index and the original Person 1/2/3 identity) correspond to a Deep SORT track carrying that same identity at that frame; instances were excluded when the tracker had reassigned a new track ID to that subject due to an identity switch, consistent with the tracker identity-switch confound documented for the Running class (Appendices Table *14*). A randomly selected 15% subset of the annotations was independently reviewed by a second annotator, yielding Cohen's kappa = 0.8765 (strong agreement) [31,32]. A second outdoor deployment video with the same camera geometry was used for cross-scene analysis. As no manual labels were available for that sequence, rule-based labels served as the deployment reference. Dataset summary and inter-annotator agreement details are provided in Appendices Tables *11*-*12*.

Parameter sensitivity analysis

Table *2* summarizes all key experimental parameters. Non-critical hyperparameters (detection threshold, Deep SORT max age, LSTM architecture) were selected by standard empirical practice. The rule-based thresholds T (loitering duration), R (loitering radius), and V (running speed threshold) directly determine operational behavior labels and pseudo-label quality, so they were examined via a dedicated two-stage sensitivity analysis. In Stage 1, a coarse sweep varied T in {60, 90, 120, 150} seconds, R in {30, 40, 50, 60, 70} pixels, and V in {100, 150, 200} pixels/second across 60 combinations, evaluating the rule-based module in isolation. Full details are provided in Appendices Table *13*. In Stage 2, a small set of

representative configurations from the viable region was evaluated under the full self-learning mode (see Table 3). The Stage 1 sweep had identified a viable region around T = 120 s and R = 50 px, with multiple running-speed thresholds achieving similar macro F1 under the pure rule-based evaluation. Building on this region, Stage 2 carried forward four representative configurations from the Stage 1 grid, plus two additional refined settings to probe intermediate radius and speed thresholds and to better separate fast walking from genuine running. The final thresholds T = 120 s, R = 50 px, and V = 125 px/s provided the best trade-off between overall accuracy and macro F1, and were adopted as the default operational settings.

Component	Configuration
Object Detection	YOLOv8-small, confidence threshold = 0.5
Tracking	Deep SORT, IoU threshold = 0.7, max age = 30
Feature Sequence	20 frames per sample
LSTM Architecture	2 layers, hidden size = 32, dropout = 0.4; multi-head attention (2 heads) with residual connection and layer norm; FC layers 32 → 32 → 16 → 3
Training	Adam, LR = 1.5e-4, weight decay = 1e-3; label smoothing = 0.1; gradient clipping (max norm = 1.0); ReduceLROnPlateau (factor = 0.5, patience = 5)
Rule-Based Running	Velocity > 125 px/s
Rule-Based Loitering	Within 50 px radius for >120 s of wall-clock processing time*
Self-Learning	Retraining every 1,200 frames; minimum buffer 300 samples; 18 or 22 fine-tuning epochs; early stopping patience = 6; Gaussian noise sigma = 0.015
Fusion Threshold	tau = 0.5 (LSTM accepted if p_max ≥ 0.5, else rule-based fallback)

TABLE 2: Key Experimental Parameters

*The loitering duration threshold T (120 s) is measured as wall-clock processing time on the reference CPU-only hardware (≈4.72 frames per second), not as native video-elapsed time. At this throughput, T = 120 s of processing time corresponds to ≈22.6 s of native video content (~566 of 2,294 frames in the 92-s deployment video). See Threshold Selection and Evaluation Independence section for full derivation.
FC, Fully Connected; IoU, Intersection over Union; LR, Learning Rate; LSTM, Long Short-Term Memory; YOLO, You Only Look Once

T (s)	R (px)	V (px/s)	Accuracy	Macro F1	F1 Loitering	F1 Running	Notes
60	60	200	0.7413	0.5076	0.7889	0.0245	

90	55	175	0.8647	0.6182	0.8896	0.0947	
90	50	150	0.8275	0.5778	0.8556	0.0482	
120	50	150	0.8718	0.651	0.8935	0.1788	
120	50	125	0.8993	0.7126	0.9213	0.3109	Best overall trade-off (adopted as default)
120	50	100	0.8845	0.7083	0.9032	0.3284	

TABLE 3: Self-Learning Sensitivity

Self-learning performance for selected threshold settings on the manually annotated deployment subset.

Evaluation metrics

Object detection was evaluated using Precision, Recall, F1, mAP@0.5, and mAP@0.5:0.95. Tracking was evaluated with MOTA, MOTP, IDF1, identity switches, and trajectory-level Precision and Recall. Behavior recognition was evaluated under two protocols: (i) manual ground-truth validation on the 4,330-sample annotated deployment subset (primary), and (ii) deployment reference evaluation against deterministic rule-based labels across the full deployment sequence (secondary).

Results And Discussion

Runtime latency

To assess runtime feasibility, the complete pipeline was profiled over a short continuous segment on the reference CPU-only hardware. The mean end-to-end latency was approximately 212 ms per frame (about 4.7 FPS), with YOLOv8-small detection dominating the cost and the combined tracking, feature, and behavior modules remaining lightweight. On this CPU-only setup, the pipeline therefore operates below real-time frame rates. Based on published YOLOv8-small inference benchmarks on comparable mid-range GPUs (e.g., NVIDIA RTX 3060-class hardware), real-time throughput near 25 FPS is plausible; however, this was not benchmarked in the present study, and GPU-based validation is identified as a required step before making real-time deployment claims (Future Work section).

Object detection and tracking

Metric	Value
Precision	89.39%
Recall	98.57%
F1 score	93.76%
mAP@0.5	0.9721

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

mAP@0.5:0.95	0.8626

TABLE 4: Detection Performance

Detection performance of YOLOv8-small on Penn-Fudan.
mAP, mean Average Precision; YOLO, You Only Look Once

Metric	Value
MOTA	74.75%
MOTP (error)	23.65%
Mean IoU	76.35%
IDF1	77.59%
ID switches	30
False positives	812
Misses	332
Precision	84.17%
Recall	92.86%
GT objects	4,650
Predicted objects	5,130
Matches	4,318

TABLE 5: Tracking Performance

Tracking performance of YOLOv8-small + Deep SORT on PETS09-S2L1.
GT, Ground Truth; IDF1, Identification F1 Score; IoU, Intersection over Union; MOTA, Multiple Object Tracking Accuracy; MOTP, Multiple Object Tracking Precision

YOLOv8-small was validated as the person detector on the Penn-Fudan Pedestrian test split (80/20 partition, 170 images) (Table *4*), and Deep SORT tracking was evaluated on PETS09-S2L1 using standard MOT metrics (Table *5*). Together, they achieved high detection precision and recall, mAP@0.5 of 0.9721, MOTA of 74.75%, IDF1 of 77.59%, and low identity-switch counts over 4,650 ground-truth instances, indicating that the YOLOv8-small + Deep SORT backbone provides sufficiently reliable and stable trajectories for downstream behavior analysis.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Behavior recognition: Manual ground-truth validation

Mode	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted F1	Samples
Self-Learning	0.8993	0.9066	0.6792	0.7126	0.8908	4,330
Supervised Avenue	0.8919	0.8814	0.6494	0.6681	0.8806	4,330
Supervised Synthetic	0.8905	0.8929	0.6751	0.7089	0.8824	4,330
Rule-Based	0.8827	0.8813	0.6623	0.6921	0.8738	4,330
Supervised Synthetic No-Fallback	0.5069	0.4407	0.4245	0.4014	0.5394	4,330
Self-Learning No-Fallback	0.5037	0.168	0.333	0.2233	0.3378	4,330

TABLE 6: Behavior Recognition

Behavior-recognition performance across modes on the 4,330-sample manually annotated deployment subset at the final thresholds (T = 120 s, R = 50 px, V = 125 px/s).

Table *6* summarizes behavior recognition performance across all modes on the 4,330-sample manually annotated deployment subset at the final thresholds. The self-learning hybrid mode achieved the strongest overall result (accuracy 0.8993, macro F1 0.7126), with supervised-synthetic and rule-based modes close behind, while both no-fallback ablations collapsed to near-random accuracy (approximately 0.50). Overall metrics across all modes are also visualized in Appendices Figure *5*.

A contextual comparison of these results against published methods from related literature is provided in the Appendices Table *18*. Because of differences in datasets and evaluation protocols, these values are not directly comparable and should be read only as illustrative reference points.

Per-class analysis in the Appendices Table *14* shows that self-learning achieves the highest Normal and Loitering F1 scores, while all modes perform worse on Running due to limited support and a short, tracker-confounded running window. Error type breakdowns and row-normalized confusion matrices (Appendices Table *15* and Figure *6*) show that most hybrid mode errors are missing Loitering events near the 120-second threshold, with false-alarm Loitering predictions remaining below 3% across rule-based and LSTM-hybrid modes, except Supervised Avenue (11.2%).

Beyond aggregate metrics, pairwise McNemar tests with continuity correction were applied to the 2,957-sample trajectory-matched subset to assess whether observed accuracy differences between modes are statistically reliable [33].

Mode	Accuracy	95% CI Lower	95% CI Upper
Self-Learning	0.9259	0.9159	0.9348

Rule-Based	0.9232	0.9131	0.9323
Supervised Avenue	0.9212	0.9109	0.9304
Supervised Synthetic	0.9192	0.9088	0.9285

TABLE 7: Accuracy

Accuracy (95% CIs) on the 2,957-sample trajectory-matched subset.

Comparison	b (SL correct, X wrong)	c (X correct, SL wrong)	Chi-square	p-value
SL vs. Rule-Based	27	19	1.0652	0.302
SL vs. Supervised Avenue	40	26	2.5606	0.11
SL vs. Supervised Synthetic	53	33	4.1977	0.041

TABLE 8: McNemar Test Results: SL vs. Other Modes

SL, Self-Learning

As summarized in Tables *7*-*8*, self-learning shows no statistically significant difference from the rule-based baseline or the Avenue-trained supervised model. Self-learning also shows a nominally lower p-value against the synthetic-only supervised model (chi-square = 4.20, p = 0.04); however, this comparison is one of three pairwise McNemar tests conducted against the same self-learning baseline (Table *8*), and p = 0.04 does not remain significant under a Bonferroni-corrected threshold of $p < 0.0167$ for three comparisons. We therefore report this result as a nominally favorable but not statistically robust difference, rather than a confirmed advantage, and note that the larger point supported by Tables *7*-*8* is that self-learning performs at least as well as all fixed-threshold alternatives without requiring manually labeled deployment data.

It should be noted that the 2,957 frame-level observations used in these McNemar comparisons originate from a small number of underlying trajectories, and adjacent frames within the same trajectory are highly correlated rather than independent. This raises a risk of pseudoreplication that may make the reported p-values overly optimistic relative to a trajectory-level or subject-level analysis. The authors treat the McNemar results here as a directional, exploratory comparison rather than a definitive statistical test, and identify trajectory-clustered or bootstrap-based inference as a necessary refinement for future evaluations with larger and more diverse deployment datasets.

Behavior recognition: Deployment reference

To evaluate alignment with deterministic rule-based logic over full deployment sequences, each learning-based mode was compared against rule-based labels across the entire primary deployment video.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Mode	Agreement	Macro Precision	Macro Recall	Macro F1	Weighted F1	Samples
Supervised Synthetic	0.9791	0.959	0.9655	0.9621	0.9791	4,635
Supervised Avenue	0.9756	0.9841	0.9062	0.9397	0.9753	4,635
Self-Learning	0.9743	0.9615	0.9621	0.9617	0.9743	4,635
Supervised Synthetic No-Fallback	0.5655	0.5116	0.6959	0.4764	0.6055	4,407
Self-Learning No-Fallback	0.5351	0.1784	0.3333	0.2324	0.373	4,407

TABLE 9: End-to-End Agreement

End-to-end agreement of learning-based modes with the rule-based deployment reference at the final thresholds.

As summarized in Table *9*, all three hybrid modes achieve very high agreement with the rule-based reference (agreement ≥ 0.974), while both no-fallback variants degrade sharply toward near-random behavior. Per-class agreement statistics, detailed deployment reference confusion matrices, and additional visual examples are provided in Appendices Table *16* and Figures *7*-*9*.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Visual and temporal analysis



FIGURE 2: Velocity-Boundary Sensitivity Case

A frame with motion near the running threshold (V approximately 5.84 px/frame; threshold = 5 px/frame, i.e., 125 px/s at 25 FPS). Rule-Based and Self-Learning classify as Running, while the supervised modes remain Normal, illustrating predictable divergence near decision boundaries.

Representative frame-level examples illustrate when the modes agree and where they diverge near decision boundaries. Figure *2* shows a velocity-boundary case near the running threshold, where the rule-based and self-learning modes predict Running while the fixed supervised models remain Normal.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

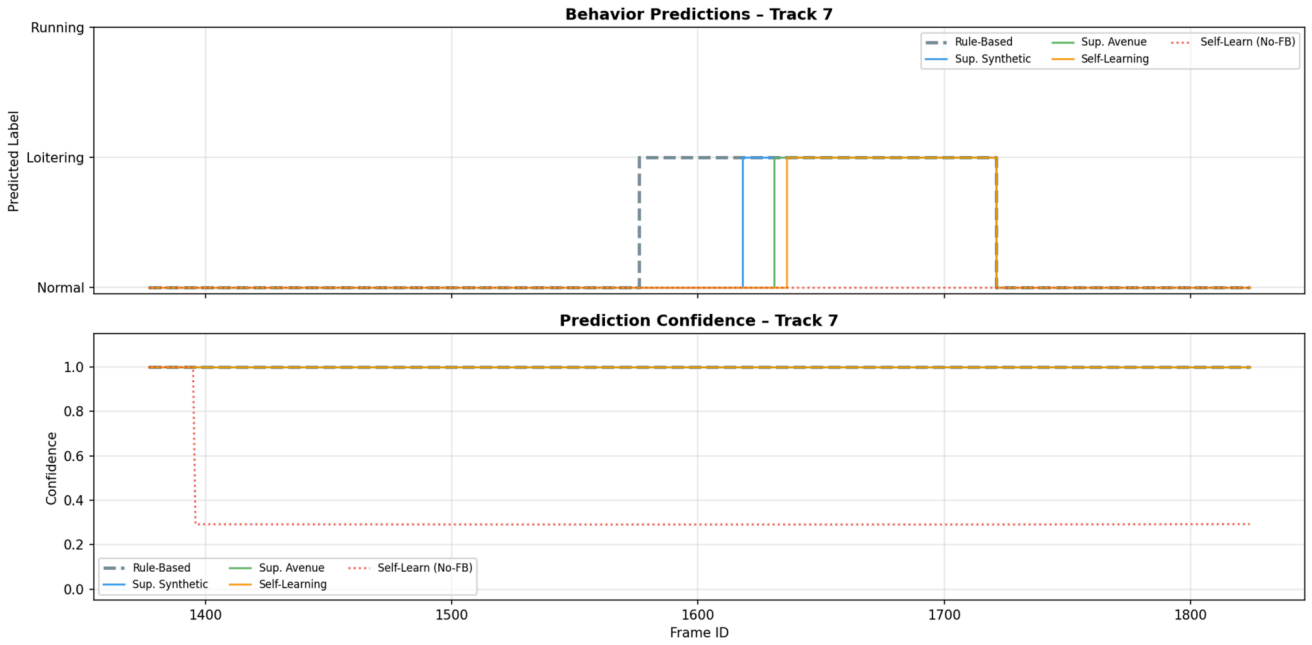


FIGURE 3: Frame-Level Prediction

Frame-level predicted labels and confidence values across operational modes. The Rule-Based module detects loitering onset earliest, followed by the supervised and Self-Learning modes in staggered order. The no-fallback ablation produces no loitering detection and remains at low confidence.

To inspect temporal behavior at the track level, frame-wise predictions and confidences were visualized across representative trajectories (Figure *3*). The rule-based module fires first once the 120-s threshold is reached, followed by the supervised and self-learning modes in staggered order, while the no-fallback variant remains almost entirely Normal with low confidence, consistent with the quantitative gaps shown earlier.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8



FIGURE 4: Second Deployment Scene

Second deployment scene used for cross-scene evaluation: a low-light outdoor location recorded from a static elevated camera with sparse pedestrian traffic.

Cross-scene generalization

To assess generalization beyond the primary deployment scene, all learning-based modes were evaluated on a second outdoor night environment sequence (Figure *4*) recorded from a different location under the same static elevated camera configuration (25 FPS, 1,920 × 1,080, 7,600 detections, 27 tracks). Since no full manual annotation was available, rule-based labels served as the deterministic deployment reference.

Mode	Accuracy	Macro F1	Normal F1	Loitering F1	Running F1
Supervised Synthetic	0.9957	0.9679	0.9973	0.9897	0.9167
Supervised Avenue	0.9888	0.9531	0.9931	0.9725	0.8936
Self-Learning	0.9845	0.9299	0.9904	0.962	0.8372
Supervised Synthetic No-Fallback	0.4663	0.3264	0.6022	0.3682	0.0087

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Self-Learning No-Fallback	0.7891	0.2984	0.8819	0.0134	0

TABLE 10: Cross-Scene Generalization

Cross-scene agreement with the rule-based reference on the second deployment video (N = 7,600 detections, 27 tracks). Values reflect agreement with deterministic rule-based labels, not validated accuracy against independent human annotation, as no manual ground truth was available for this scene.

Table *10* shows that the three hybrid modes maintain strong agreement with the rule-based reference in this new scene: supervised synthetic achieves 0.9957 agreement with the rule-based reference (agreement macro F1 0.9679), supervised Avenue achieves 0.9888 agreement (0.9531), and self-learning achieves 0.9845 agreement (0.9299). No-fallback variants again degrade sharply, particularly for Loitering and Running, mirroring the pattern observed on the primary deployment sequence.

Discussion

Threshold Selection and Evaluation Independence

The two-stage sensitivity analysis identified T = 120 s, R = 50 px, and V = 125 px/s as effective operating thresholds for controlled surveillance. Moving from T = 90 s to T = 120 s at fixed R and V improved accuracy and macro F1 and more than doubled Running F1, indicating that longer loitering accumulation cleans pseudo-labels for LSTM adaptation. The final setting, T = 120 s, was calibrated as wall-clock processing time on the reference CPU-only hardware, corresponding to approximately 22.6 s of native video content at the measured throughput of 4.72 FPS. Because independent manual ground-truth annotations were produced by human annotators judging the footage directly, without reference to the system's internal timer, the quantity actually validated against human judgment is this approximately 22.6 s effective video-content dwell time rather than the nominal 120 s wall-clock value. The authors do not interpret T = 120 s as representing a literal two-minute operational standard; the effective threshold of approximately 22.6 s is a plausible, human-consistent dwell criterion for the short evaluation clips used here, but it is a hardware-calibration artifact rather than a deliberately chosen security-policy duration. Consequences for real-time GPU deployment, where this hardware dependence does not hold, are discussed further below.

The authors note an important limitation regarding the relationship between threshold calibration and primary evaluation. The final operating thresholds (T = 120 s, R = 50 px, V = 125 px/s) were selected in Stage 2 of the sensitivity analysis using performance measured on the same manually annotated deployment subset subsequently used for the primary behavior-recognition comparison (Table *6*). This shared-data design introduces a risk of optimistic bias in the reported accuracy and macro F1 values, since the thresholds are, by construction, tuned to this specific subset rather than validated on held-out data. This limitation does not affect the self-learning mode's core claim of requiring no manually labeled data for LSTM training. The LSTM itself is fine-tuned only on rule-derived pseudo-labels, but this means that the rule-based thresholds underlying that pseudo-labeling process were indirectly informed by manual annotations during calibration. A more rigorous evaluation protocol would use an independent validation split for threshold selection and a fully held-out test set for final performance reporting, or a cross-validation scheme across multiple deployment sequences. The authors identify this as a priority for future work (Future Work section, first priority).

Self-Learning and Annotation

Under independent manual-ground-truth validation, self-learning shows no statistically significant difference from the rule-based baseline and shows a nominally lower p-value against the synthetic-only supervised model, an effect that does not survive Bonferroni correction across the three pairwise comparisons (see Behavior Recognition: Manual Ground-Truth Validation section). The absolute accuracy margin over the rule-based mode is modest but important when interpreted against annotation cost. The rule-based configuration requires expert tuning of T, R, and V using labeled

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

deployment data, whereas self-learning reaches a slightly better operating point without any manually labeled samples from the deployment scene. The triangular disagreement analysis provided in the Appendices Table *17* shows that the LSTM does not redefine loitering but learns a spatio-temporal signature of pre-threshold loitering and produces confident loitering predictions as this signature accumulates in frames approaching the threshold boundary.

Hybrid Fallback Design

The hybrid fallback design is functionally necessary. The attention-enhanced LSTM operating on 20-frame sequences provides capabilities that deterministic thresholds alone do not offer, such as aggregating evidence over multiple frames, emphasizing discriminative temporal segments, and smoothing brittle single-frame threshold errors near velocity boundaries. At the same time, LSTM-only operation is clearly insufficient: both no-fallback variants collapse toward near-random accuracy on the deployment sequence, and their degradation relative to the full hybrid configuration is severe. The rule-based and learning-based components therefore play complementary roles, with the rule-based path ensuring interpretable behavior and providing a safe fallback, while the LSTM refines decisions near boundaries by reducing sensitivity to noisy instantaneous measurements.

Scope of Ablation

The present ablation design isolates the contribution of the classification strategy (rule-based vs. supervised vs. self-learning, with and without fallback fusion) while holding the detection-tracking-feature backbone constant across all six modes, since this backbone was validated independently on public benchmarks (Penn-Fudan for detection, Table *4*, and PETS09-S2L1 for tracking, Table *5*) prior to deployment. This design was chosen because the primary research question concerns mode selection and fallback necessity, not backbone architecture search. A finer-grained ablation isolating the individual contributions of the detection confidence threshold, tracking parameters (IoU threshold and max age), or feature-vector dimensionality was outside the scope of this study and is noted as a direction for future work.

Pixel Threshold Transfer

All operational thresholds are defined in pixel space, which ties them to the camera geometry and scene layout. In the present deployment, a static elevated camera with a fixed field of view reduces depth variation and keeps pixel-based statistics consistent across subject positions. In more variable configurations, the same pixel threshold corresponds to different real-world distances and speeds depending on subject depth and perspective. Adding a perspective correction stage that maps trajectories to a ground plane or bird's-eye view would allow behavioral thresholds to be expressed in metric units and reused across sites with different mounting heights and viewing angles, thereby improving transferability across installations.

Limitations

Several limitations should be considered when interpreting the results. First, the primary and cross-scene deployment videos were recorded using a single camera geometry (static, elevated) with a small number of subjects (three in the primary scene), which constrains the diversity of pose, scale, and occlusion patterns represented in the manually annotated evaluation; results on more heterogeneous camera placements or larger subject populations may differ. Second, both deployment videos were recorded by the same research team under similar environmental conditions (indoor and outdoor night settings within a single institutional context), which may introduce dataset bias not representative of the full range of controlled surveillance environments (e.g., varying lighting, weather, or crowd density). Third, computational complexity was assessed only on a single CPU-only reference platform (Computing Environment section); while parameter counts for the LSTM module are modest (two 32-unit LSTM layers plus a lightweight attention block), end-to-end latency is dominated by YOLOv8-small inference, and formal complexity analysis (FLOPs, memory footprint across hardware tiers) was not performed. These limitations motivate the multi-scene, cross-hardware evaluation identified in Future Work.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Future work

Future work will focus on four priorities. First, multi-scene evaluation across additional controlled environments with varied camera configurations and independent annotations is needed to quantify generalization. Second, comparison with published loitering detection and video anomaly detection baselines on the same deployment data will situate the proposed framework more precisely within the broader literature. Third, improved Running class evaluation requires deployment data with temporally distributed running episodes under stable track identities, followed by class-balanced retraining and threshold recalibration. Finally, incorporating a ground plane or bird's-eye-view projection would convert pixel-based thresholds into metric units, decoupling operational settings from camera geometry and enabling easier deployment across heterogeneous installations.

Conclusions

This paper presented a multi-mode framework for suspicious behavior detection in controlled surveillance environments, integrating YOLOv8-based person detection, Deep SORT multi-object tracking, spatio-temporal feature extraction, and a hybrid decision pipeline that combines rule-based and LSTM-based modules with supervised and self-learning operating modes. A two-stage sensitivity analysis identified T = 120 s, R = 50 px, and V = 125 px/s as operationally effective thresholds. Under independent manual-ground-truth validation at this operating point, the self-learning hybrid mode achieved the highest accuracy (0.899) and macro F1 (0.713) among the six evaluated modes without manually labeled data used for LSTM training (Table *6*), although the underlying rule-based thresholds were calibrated using the same evaluation subset (see Threshold Selection and Evaluation Independence), while purely supervised and purely rule-based modes remained within 0.02 accuracy of this result. Removing the fallback fusion caused both LSTM-only ablations to degrade toward near-random behavior, confirming that fallback is a functionally essential component. Running class recognition remained a limitation due to weak class support and track identity reassignment in the available recordings.

Preliminary cross-scene evaluation on a second outdoor night deployment video showed that the hybrid modes retain high agreement with the rule-based reference and reproduce the same no-fallback degradation, providing initial evidence that the multi-mode design and threshold calibration can transfer beyond a single scene, although full multi-scene validation with independent manual annotations remains open.

Appendices

Derivation of the sequence length N

This section derives the feasible range for the sequence length N used in the spatio-temporal descriptor and justifies the choice of N = 20 frames used in the Materials and Methods section of the main paper.

Running detection requires at least 3 consecutive high-velocity frames to avoid spurious one-frame spikes. Applying a warm-up factor of 2 for LSTM hidden-state stabilization gives a lower bound:

$$N_{\text{min}} = 2 \times 3 = 6 \text{ frames}$$

Loitering onset should be detected within a response time of 0.5 seconds, which, at a frame rate of 25 FPS, yields:

$$N_{\text{loiter}} = \lceil 25 \times 0.5 \rceil = 13 \text{ frames}$$

From the deployment perspective, an upper bound on first-prediction latency of 5 seconds is imposed. At the measured CPU-only throughput of 4.72 FPS, this gives:

$$N_{\text{max}} = \lfloor 4.72 \times 5 \rfloor = 23 \text{ frames}$$

Combining these constraints:

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

$$\max(N_{\min}, N_{\text{loiter}}) \leq N \leq N_{\max}$$

or numerically,

$$13 \leq N \leq 23$$

Within this range, N = 20 frames was selected as a value within this range that simultaneously satisfies running-detection, loitering-response, and latency constraints.

Dataset summary

Table 11 provides a compact summary of all datasets and video sources used in the proposed framework. The roles of each source are described in detail in the Materials and Methods section of the main paper.

Dataset	Data Type	Primary Use	Stage/Mode	Notes
Penn-Fudan	Annotated pedestrian images	Baseline person detection evaluation	Detection	Used to verify that the pretrained YOLOv8s detector provides sufficiently reliable pedestrian localization for downstream tracking; 170 images; 80/20 train/test split.
PETS09-S2L1	Annotated surveillance video	Multi-object tracking evaluation	Tracking evaluation	Elevated static camera; ≤6 persons/frame; closely approximates restricted-area monitoring conditions.
CUHK Avenue	Annotated surveillance videos	Supervised behavior pre-training only	Supervised	Public benchmark containing normal and abnormal events (running, loitering) relevant to trajectory-based analysis.
Synthetic Dataset	Simulated behavior sequences	Controlled supervised training for Normal, Loitering, and Running	Supervised	4,700 samples (2,000 Normal; 1,500 Loitering; 1,200 Running); generated from class-specific Gaussian distributions parameterized to reflect rule-based kinematic constraints.

Deployment Video 1	Raw surveillance video	Pseudo-label generation, retraining, deployment evaluation	Self-learning / validation	Single indoor sequence; 25 frames per second, 1,920 × 1,080 px, ~92 s (2,294 frames) (see Rule-Based Loitering threshold note in Table 2); 3 unique subjects (Normal, Loitering, and Running).
Deployment Annotated Subset	Frame-level annotated samples	Independent end-to-end validation against human labels	Validation	Produced in Label Studio; 4,392 labeled frame-person instances; 4,330 valid matched samples after trajectory alignment (2,183 Normal; 1,987 Loitering; 160 Running).
Second Deployment Video	Raw surveillance video	Cross-scene generalization evaluation	Cross-scene	Outdoor sequence; 25 a { text-decoration: none; color: #464feb; } tr th, tr td { border: 1px solid #e6e6e6; } tr th { background-color: #f5f5f5; } frames per second , 1,920 × 1,080 px; 7,600 frame-level detections; 27 unique tracks; rule-based labels used as deployment reference.

TABLE 11: Dataset Summary

Datasets and deployment video sources used in the proposed framework.

Annotation quality, inter-annotator agreement

The complete inter-annotator agreement is reported below, as referenced in the Materials and Methods section of the main paper.

Metric	Value	Interpretation

Subset size (frame-person instances)	658	15% of the 4,392 annotated instances
Cohen's kappa	0.8765	Almost perfect agreement
Landis & Koch band	0.81-1.00	Almost perfect

TABLE 12: Inter-Annotator Agreement

Inter-annotator agreement on the 15% independent re-annotation subset.

The almost perfect inter-annotator agreement (kappa = 0.8765) confirms that the annotation protocol is reproducible and that the manual ground-truth labels are reliable for downstream evaluation.

Annotator independence from system timing

Annotators labeled Normal, Loitering, and Running directly from the video footage in Label Studio, based on their own visual judgment of subject motion and dwell behavior. Annotators had no access to the rule-based module's internal wall-clock timer or to any system output during labeling. Consequently, the manual ground-truth labels constitute an evaluation standard that is independent of the wall-clock timing basis discussed under Threshold Selection and Evaluation Independence in the Results and Discussion section, and the resulting agreement in Table *6* reflects genuine correspondence between the rule's effective ~22.6 s video-content dwell criterion and human visual judgment of loitering behavior, not circular validation against the system's own internal clock.

Sensitivity analysis - Full Stage-1 sweep

The complete Stage 1 rule-based sensitivity sweep is reported in Table *13*. The two-stage sensitivity analysis is summarized in the main paper's Materials and Methods section, where Stage 2 carried forward representative configurations to full end-to-end self-learning evaluation.

T (s)	R (px)	V (px/s)	Precision	Recall	Macro F1	Accuracy	Stage 2 Representative
60	30	100	0.5102	0.4846	0.2423	0.3346	
60	30	150	0.5265	0.4786	0.2746	0.3931	
60	30	200	0.5515	0.4968	0.3118	0.4412	
60	40	100	0.5117	0.4901	0.2537	0.3422	
60	40	150	0.528	0.4841	0.2861	0.4007	
60	40	200	0.5531	0.5023	0.3234	0.4488	
60	50	100	0.5153	0.5047	0.2825	0.3623	
60	50	150	0.532	0.4987	0.3151	0.4208	
60	50	200	0.5573	0.5169	0.3526	0.4689	

60	60	100	0.5168	0.5097	0.2919	0.3692	
60	60	150	0.5334	0.5037	0.3246	0.4277	
60	60	200	0.5587	0.5219	0.3621	0.4758	Yes
60	70	100	0.5168	0.5097	0.2919	0.3692	
60	70	150	0.5334	0.5037	0.3246	0.4277	
60	70	200	0.5587	0.5219	0.3621	0.4758	
90	30	100	0.1745	0.475	0.2218	0.3214	
90	30	150	0.1907	0.469	0.254	0.3799	
90	30	200	0.2158	0.4872	0.2912	0.428	
90	40	100	0.1745	0.475	0.2218	0.3214	
90	40	150	0.1907	0.469	0.254	0.3799	
90	40	200	0.2158	0.4872	0.2912	0.428	
90	50	100	0.1745	0.475	0.2218	0.3214	
90	50	150	0.1907	0.469	0.254	0.3799	Yes
90	50	200	0.2158	0.4872	0.2912	0.428	
90	60	100	0.1745	0.475	0.2218	0.3214	
90	60	150	0.1907	0.469	0.254	0.3799	
90	60	200	0.2158	0.4872	0.2912	0.428	
90	70	100	0.1745	0.475	0.2218	0.3214	
90	70	150	0.1907	0.469	0.254	0.3799	
90	70	200	0.2158	0.4872	0.2912	0.428	
120	30	100	0.1745	0.475	0.2218	0.3214	
120	30	150	0.1907	0.469	0.254	0.3799	
120	30	200	0.2158	0.4872	0.2912	0.428	
120	40	100	0.1745	0.475	0.2218	0.3214	
120	40	150	0.1907	0.469	0.254	0.3799	

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

120	40	200	0.2158	0.4872	0.2912	0.428	
120	50	100	0.1745	0.475	0.2218	0.3214	Yes
120	50	150	0.1907	0.469	0.254	0.3799	Yes
120	50	200	0.2158	0.4872	0.2912	0.428	
120	60	100	0.1745	0.475	0.2218	0.3214	
120	60	150	0.1907	0.469	0.254	0.3799	
120	60	200	0.2158	0.4872	0.2912	0.428	
120	70	100	0.1745	0.475	0.2218	0.3214	
120	70	150	0.1907	0.469	0.254	0.3799	
120	70	200	0.2158	0.4872	0.2912	0.428	
150	30	100	0.1745	0.475	0.2218	0.3214	
150	30	150	0.1907	0.469	0.254	0.3799	
150	30	200	0.2158	0.4872	0.2912	0.428	
150	40	100	0.1745	0.475	0.2218	0.3214	
150	40	150	0.1907	0.469	0.254	0.3799	
150	40	200	0.2158	0.4872	0.2912	0.428	
150	50	100	0.1745	0.475	0.2218	0.3214	
150	50	150	0.1907	0.469	0.254	0.3799	
150	50	200	0.2158	0.4872	0.2912	0.428	
150	60	100	0.1745	0.475	0.2218	0.3214	
150	60	150	0.1907	0.469	0.254	0.3799	
150	60	200	0.2158	0.4872	0.2912	0.428	
150	70	100	0.1745	0.475	0.2218	0.3214	
150	70	150	0.1907	0.469	0.254	0.3799	
150	70	200	0.2158	0.4872	0.2912	0.428	

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

TABLE 13: Stage-1 Sensitivity Sweep

Stage-1 boundary exclusions: configurations with T < 60 s produced near-zero Running F1 (all loitering events occurred within the first 74 s of wall-clock processing time, corresponding to approximately 14 s of native video content at the measured 4.72 frames per second throughput described under Runtime Latency subsection in the Results and Discussion section), and V > 200 px/s produced no Running detections. These boundary regions were excluded from Stage 2.

Per-class F1-scores, manual ground-truth evaluation

Table *14* reports per-class F1 scores (referenced in the Results and Discussion section of the main paper) for all six operational modes on the 4,330-sample manually annotated deployment subset at the final threshold setting (T = 120 s, R = 50 px, V = 125 px/s).

Note on Running-class results: All 145 Running-class annotations are concentrated in a single 69-frame window (frames 2,184-2,252), after Deep SORT had reassigned subjects from primary track IDs (1, 2, 3) to secondary IDs (8, 10, 13). This constitutes a tracker identity-switch confound, a known failure mode of appearance-based trackers under detection-gap conditions, which prevents reliable cross-mode Running F1 comparison. Normal and Loitering results constitute the primary evidence base.

Mode	Normal F1	Loitering F1	Running F1
Self-Learning	0.9055	0.9213	0.3109
Supervised Synthetic	0.8972	0.9115	0.3179
Supervised Avenue	0.8982	0.9171	0.1889
Rule-Based	0.8887	0.905	0.2827
Supervised Synthetic No-Fallback	0.5369	0.5785	0.0887
Self-Learning No-Fallback	0.6699	0	0

TABLE 14: Per-Class F1 on Manual Ground Truth

Per-class F1 scores on the manually annotated deployment subset at the final threshold setting (T = 120 s, R = 50 px, V = 125 px/s). Running-class results are indicative only (see note above).

Error-type breakdown

Table *15* provides the full error-type breakdown (referenced in the Results and Discussion section of the main paper) on the 2,957-sample trajectory-confirmed deployment subset. "Missed Loitering" is defined as predicting Normal when the ground truth is Loitering, corresponding to detection lag at the 120-s accumulation boundary (wall-clock; ≈22.6 s video-equivalent). "False Alarm" is defined as predicting Loitering when the ground truth is Normal.

Mode	Missed Loitering	Missed Loitering %	False Alarm	False Alarm %
Rule-Based	220	96.9	6	2.6

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

Self-Learning	213	97.3	5	2.3
Supervised Synthetic	231	96.7	7	2.9
Supervised Avenue	206	88.4	26	11.2

TABLE 15: Error Type Breakdown

Error type breakdown on the manually annotated deployment subset (N = 2,957 confirmed matched frames).

Missed Loitering accounts for at least 88.4% of all errors in every hybrid mode, confirming that the dominant failure mode is a detection lag at the loitering onset boundary, not classifier confusion. False Alarm rates remain below 3% for rule-based and all LSTM-hybrid modes except Supervised Avenue (11.2%), which is operationally important in controlled environments where false alarms cause alert fatigue.

Per-class F1 scores, deployment-reference evaluation

Table *16* reports per-class F1 scores for the end-to-end deployment reference evaluation (rule-based labels as reference) at the final threshold setting, as referenced in the Results and Discussion section of the main paper.

Mode	Normal F1	Loitering F1	Running F1
Supervised Synthetic	0.9814	0.9787	0.9261
Supervised Avenue	0.9783	0.9771	0.8636
Self-Learning	0.9772	0.9725	0.9353
Supervised Synthetic No-Fallback	0.5863	0.6492	0.1937
Self-Learning No-Fallback	0.6971	0	0

TABLE 16: Per-Class F1 Deployment Reference

Per-class F1 scores for end-to-end behavior-recognition agreement against the rule-based deployment reference at the final threshold setting (T = 120 s, R = 50 px, V = 125 px/s).

Triangular disagreement analysis

Table *17* presents the triangular disagreement analysis (referenced in the Discussion section of the main paper), comparing the Rule-Based module, the Self-Learning hybrid mode, and independent Manual Ground Truth on the 2,957-sample trajectory-confirmed deployment subset.

Category	Frames	Percent
Both Rule-Based and Self-Learning correct	2,711	91.7

Both wrong, systematic detection lag	200	6.8
Self-Learning correct, Rule-Based wrong	27	0.9
Rule-Based correct, Self-Learning wrong	19	0.6
Total	2,957	100

TABLE 17: Triangular Disagreement Analysis

Triangular disagreement analysis: Rule-Based vs. Self-Learning vs. Manual Ground Truth (N = 2,957 frames). The 6.8% shared-error category corresponds to frames within the 120-s Loitering accumulation window (wall-clock; ≈22.6 s video-equivalent) and represents systematic detection lag, not mode-specific confusion.

Interpretation: The 27 frames (0.9%) where Self-Learning is correct and Rule-Based is wrong reflect LSTM pattern generalization near the loitering onset boundary - the network recognizes the 20-frame spatio-temporal signature of pre-threshold loitering motion and produces a confident Loitering classification slightly before the 120-s mark, without redefining the loitering criterion.

Contextual comparison with related work

Table *18* situates the primary behavior-recognition results (Table *6*) alongside published findings from closely related work. Because these studies use different datasets, camera configurations, and label protocols, the values below should be read as contextual reference points illustrating the general performance range reported in this literature, not as a controlled head-to-head comparison. A direct benchmark on shared deployment data is identified as a priority for future work (Future Work section, second priority).

Study	Method	Reported Metric	Dataset
This work	Rule + LSTM hybrid (self-learning)	Accuracy 0.899, Macro F1 0.713	Own deployment video (controlled)
Wahyono et al. [9]	ViBe + Random Forest	Accuracy 0.850 (85.0%)	Custom loitering dataset
Nasir et al. [21]	CADF (YOLOv8 + Soft-NMS)	Accuracy/AUC 0.996 (Precision 0.982, Recall 0.991, F1 0.986)	UCSD Ped2 (public)
Elnady and Abdelmunim [12]	YOLOv7 + Deep SORT + LSTM	Average Accuracy 0.9858 (98.58%)	UCF101, KTH, WEIZMANN, IXMAS
Kim et al. [20]	YOLOv4 + KCF tracking	Average Accuracy 0.9465 (94.65%)	KISA datasets

TABLE 18: Contextual Comparison

Contextual comparison with published results from related literature. Values are not directly comparable due to differing datasets and evaluation protocols; see note above.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

AUC, Area Under the Curve; CADF, Crowd Anomaly Detection Framework; IXMAS, INRIA Xmas Motion Acquisition Sequences; KCF, Kernelized Correlation Filter; KISA, Korea Internet & Security Agency; KTH, Kungliga Tekniska Högskolan (Royal Institute of Technology); LSTM, Long Short-Term Memory; NMS, Non-Maximum Suppression; SORT, Simple Online and Realtime Tracking; UCF101, University of Central Florida 101-action dataset; UCSD Ped2, University of California San Diego Pedestrian dataset 2; ViBe, Visual Background Extractor; YOLO, You Only Look Once

Figure *5* summarizes overall accuracy, macro precision, macro recall, and macro F1 for all operational modes on the manually annotated deployment subset.

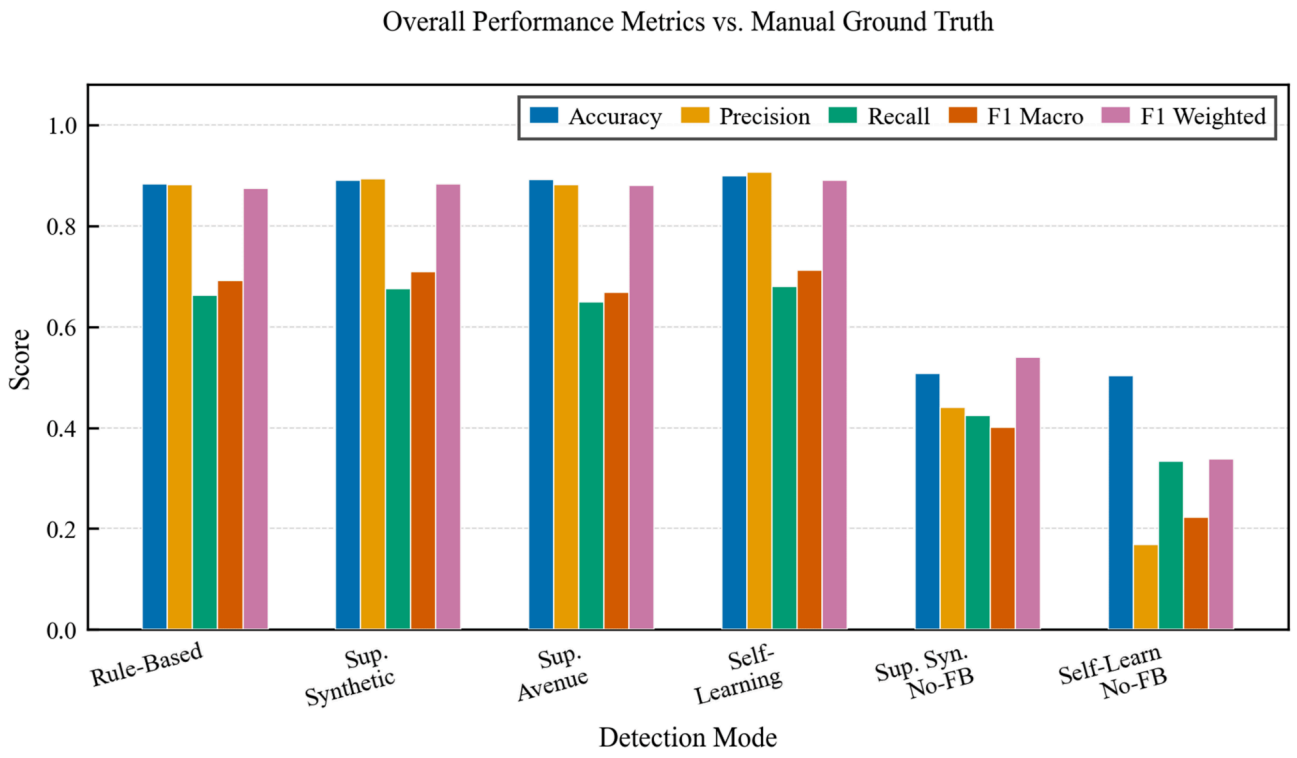


FIGURE 5: Overall Performance Matrices

Overall performance metrics across all operational modes on the manually annotated deployment subset at the final threshold setting (T = 120 s, R = 50 px, V = 125 px/s).

Figure *6* shows the row-normalized confusion matrices on the manually annotated deployment subset; most errors are Loitering frames predicted as Normal near the loitering threshold boundary.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

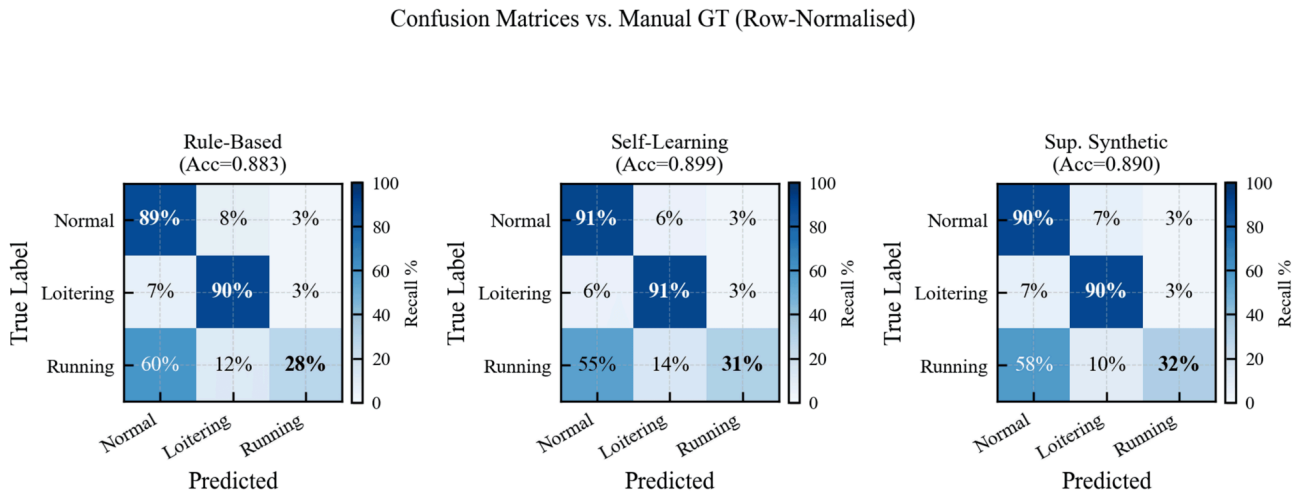


FIGURE 6: Confusion Matrices - Three Representative Modes (Manual Ground Truth)

Row-normalized confusion matrices on the manually annotated deployment subset for the rule-based mode, the self-learning hybrid mode, and the strongest fixed supervised baseline.

Figure *7* shows the confusion matrices under the deployment-reference evaluation, with rule-based labels as the reference.

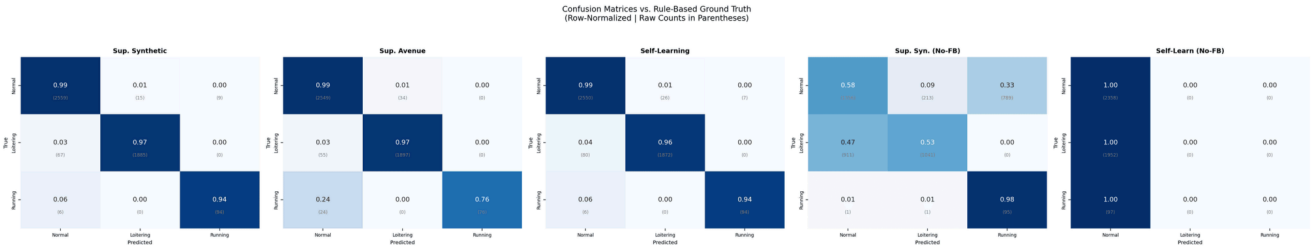


FIGURE 7: Confusion Matrices - Five Learning-Based Modes (Deployment Reference)

Confusion matrices for the five learning-based modes under the deployment-reference evaluation at the final threshold setting.

Figure *8* compares overall and per-class performance under the deployment-reference evaluation; the no-fallback variants degrade sharply relative to the hybrid modes.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

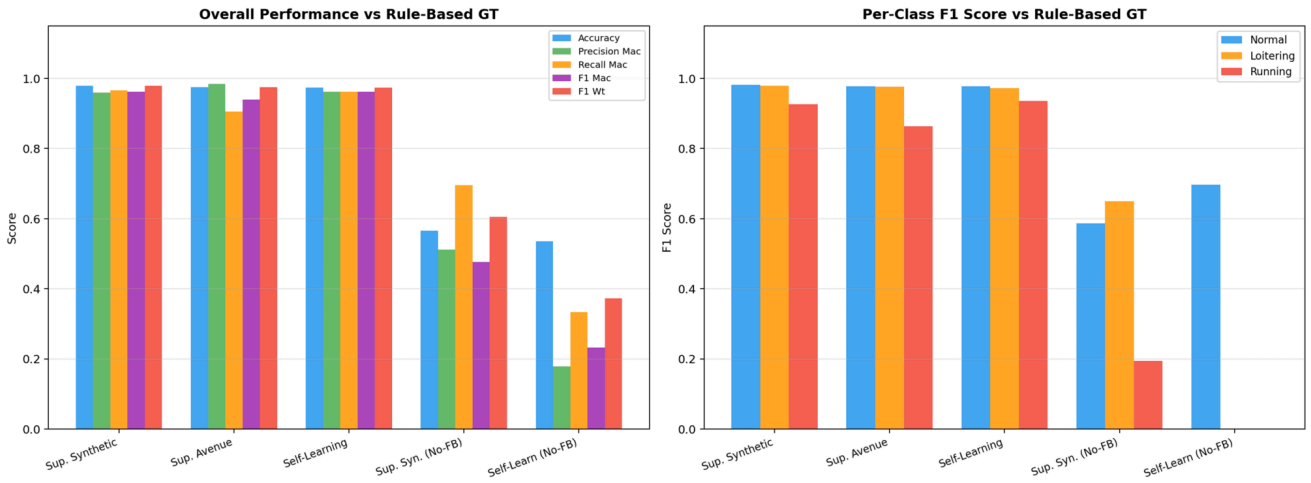


FIGURE 8: Metrics Comparison - Deployment Reference

Overall and per-class performance of the five learning-based modes under the deployment-reference evaluation at the final threshold setting.

Figure *9* shows the confidence analysis under the deployment-reference setting; hybrid modes and the rule-based baseline maintain higher and more stable confidence than the no-fallback variants.

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

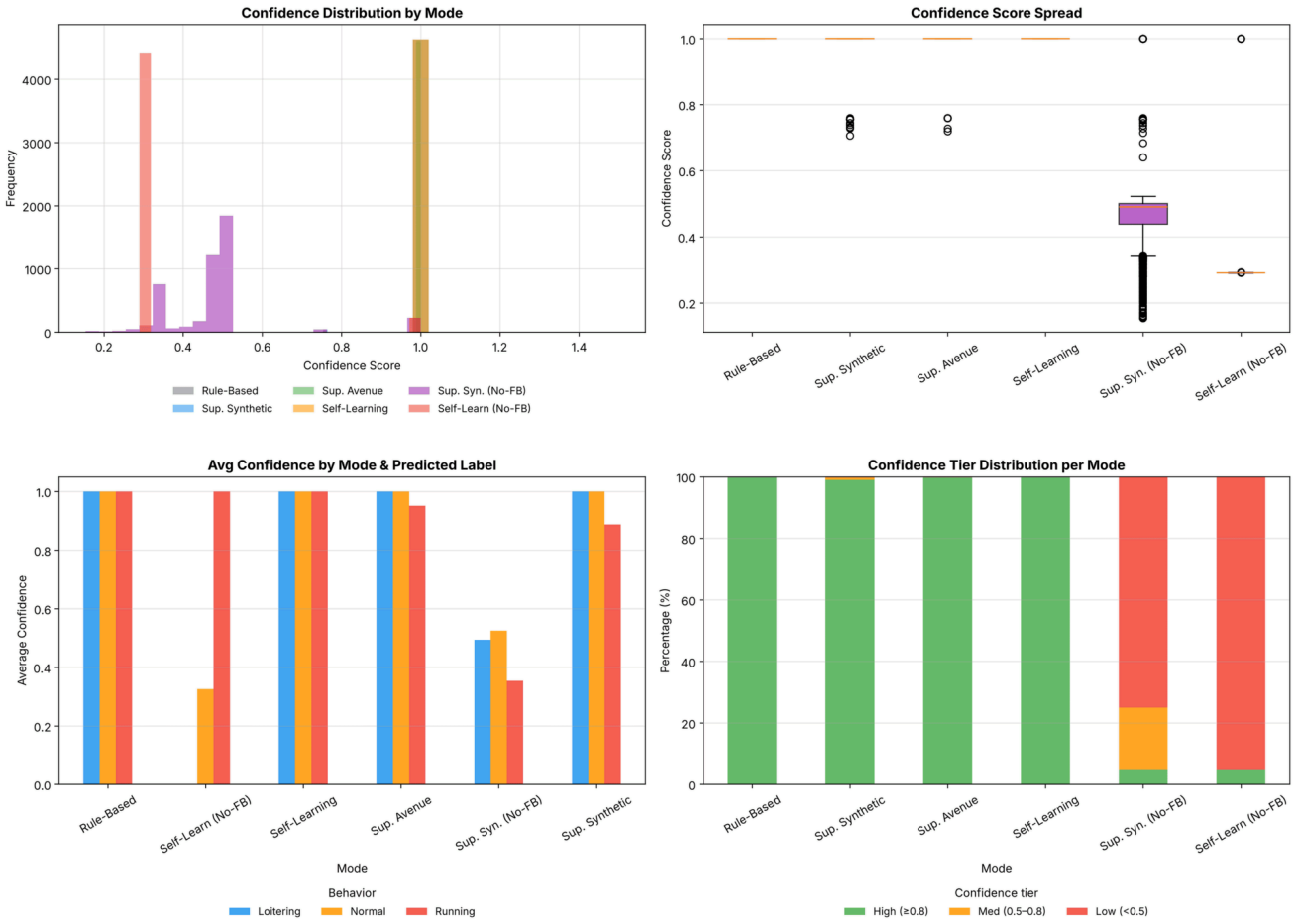


FIGURE 9: Confidence Analysis

Confidence analysis across operational modes under the deployment-reference setting, including overall confidence distributions and average confidence by predicted label. Hybrid modes and the Rule-Based baseline exhibit higher and more stable confidence than no-fallback variants.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Rabbi Mushad, Md Akhtaruzzaman

Acquisition, analysis, or interpretation of data: Rabbi Mushad

Drafting of the manuscript: Rabbi Mushad

Critical review of the manuscript for important intellectual content: Md Akhtaruzzaman

Supervision: Md Akhtaruzzaman

Disclosures

Human subjects: Consent was obtained from all participants in this study. Not applicable. issued approval Not applicable. This study involved video recordings of a consenting adult volunteer, recruited through the authors' professional network, for the purpose of evaluating a computer-vision-based behavior-detection framework.

Written informed consent for recording, analysis, and publication of derived data and images was obtained prior to participation. All identifiable facial information has been anonymized (blurred) in every published figure. This study constitutes non-invasive, minimal-risk observational research rather than clinical, behavioral, or biomedical human-subjects research; no institutional ethics committee at the authors' affiliated institutions (Military Institute of Science and Technology; Daffodil International University) currently mandates formal review for this category of study, and no formal committee approval or reference number was therefore sought. The work was conducted in accordance with the ethical principles of the Declaration of Helsinki regarding voluntary participation, consent, privacy, and data protection. Signed consent documentation is retained by the corresponding author and is available upon reasonable request.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The datasets (and/or code) supporting this study are available from the corresponding author upon reasonable request.

Acknowledgements

Data are available on reasonable request. The deployment surveillance datasets generated and/or analyzed during this study are not publicly available due to operational security restrictions, but are available from the corresponding author (Rabbi Mushad, mushad9822@gmail.com) on reasonable request. The public benchmark datasets used (Penn-Fudan, PETS09-S2L1, CUHK Avenue) are available from their original sources.

References

1. Amrutha CV, Jyotsna C, Amudha J: Deep learning approach for suspicious activity detection from surveillance video. 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India. 2020, 335-339. 10.1109/ICIMIA48430.2020.9074920
2. Duong HT, Le VT, Hoang VT: Deep learning-based anomaly detection in video surveillance: a survey. Sensors. 2023, 23:5024. 10.3390/s23115024
3. Kwak N, Lee B: Deep learning applied abnormal human behavior detection in video surveillance systems - a survey. International Journal of Contents. 2024, 20:84-95. 10.5392/ijoc.2024.20.4.084
4. Liu J, Liu Y, Lin J, et al.: Networking systems for video anomaly detection: a tutorial and survey. ACM Computing Surveys. 2025, 57:1-37. 10.1145/3729222
5. Siva P, Pujitha GB, Krishna GS, Hemanth G, Sai Teja BM: Smart surveillance systems using YOLOv8: a scalable approach for crowd and threat detection. International Journal of Recent Advances in Engineering and Technology. 2025, 14:44-53. 10.65521/intjournalrecadvengtech.v14i1.178
6. Krishnan SR, Amudha P, Sivakumari S: Real-time abnormal behaviour detection using energy-efficient YOLO-based framework. Scientific Reports. 2025, 15:38967. 10.1038/s41598-025-22888-1
7. Qasim M, Verdu E: Video anomaly detection system using deep convolutional and recurrent models. Results in Engineering. 2023, 18:101026. 10.1016/j.rineng.2023.101026

8. Chandrakala S, Deepak K, Revathy G: Anomaly detection in surveillance videos: a thematic taxonomy of deep models, review and performance analysis. Artificial Intelligence Review. 2023, 56:3319-3368. 10.1007/s10462-022-10258-6

9. Wahyono, Harjoko A, Dharmawan A, Adhinata FD, Kosala G, Jo K-H: Loitering detection using spatial-temporal information for intelligent surveillance systems on a vision sensor. Journal of Sensor and Actuator Networks. 2023, 12:9. 10.3390/jsan12010009

10. Ben Ayed M, Elkosantini S, Alshaya SA, Abid M: Suspicious behavior recognition based on face features. IEEE Access. 2019, 7:149952-149958. 10.1109/access.2019.2947338

11. Gupta N, Agarwal BB: Recognition of suspicious human activity in video surveillance: a review. Engineering, Technology & Applied Science Research. 2023, 13:10529-10534. 10.48084/etasr.5739

12. Elnady M, Abdelmunim HE: A novel YOLO LSTM approach for enhanced human action recognition in video sequences. Scientific Reports. 2025, 15:17036. 10.1038/s41598-025-01898-z

13. Mohamed Zaidi M, Avelino Sampedro G, Almadhor A, et al.: Suspicious human activity recognition from surveillance videos using deep learning. IEEE Access. 2024, 12:105497-105510. 10.1109/access.2024.3436653

14. Zhu L, Wang L, Raj A, Gedeon T, Chen C: Advancing video anomaly detection: a concise review and a new dataset. Advances in Neural Information Processing Systems 37. 2024, 89943-89977. 10.52202/079017-2856

15. Sreenu G, Saleem Durai MA: Intelligent video surveillance: a review through deep learning techniques for crowd analysis. Journal of Big Data. 2019, 6:48. 10.1186/s40537-019-0212-5

16. Abdalla M, Javed S, Al Radi M, Ulhaq A, Werghi N: Video anomaly detection in 10 years: a survey and outlook. Neural Computing and Applications. 2025, 37:26321-26364. 10.1007/s00521-025-11659-8

17. Ibrahim AA, Abosamra G, Dahab M: Real-time anomalous behavior detection of students in examination rooms using neural networks and Gaussian distribution. International Journal of Scientific and Engineering Research. 2018, 9:1716-1724.

18. Auti S, Darade P, Munje A, Patil N, Khushwaha A: Deep learning approach for suspicious activity detection from surveillance video in examination hall. International Journal of Advanced Research in Science Communication and Technology. 2025, 5:627-635. 10.48175/ijarsct-23073

19. Pathak AR, Pandey M, Rautaray S: Application of deep learning for object detection. Procedia Computer Science. 2018, 132:1706-1717. 10.1016/j.procs.2018.05.144

20. Kim D, Kim H, Mok Y, Paik J: Real-time surveillance system for analyzing abnormal behavior of pedestrians. Applied Sciences. 2021, 11:6153. 10.3390/app11136153

21. Nasir R, Jalil Z, Nasir M, Alsubait T, Ashraf M, Saleem S: An enhanced framework for real-time dense crowd abnormal behavior detection using YOLOv8. Artificial Intelligence Review. 2025, 58:202. 10.1007/s10462-025-11206-w

22. Edozie E, Shuaibu AN, John UK, Sadiq BO: Detection of anomalous activities around telecommunications infrastructure based on YOLOv8s. Scientific Reports. 2025, 15:38805. 10.1038/s41598-025-22680-1

23. Poonia RC, Upreti K, Singh N, Kesarwani J, Alam MS: An efficient detection of suspicious objects from dynamic video surveillance by fusion-based multiview deep learning techniques. Journal of Mobile Multimedia. 2025, 21:1-26. 10.13052/jmm1550-4646.2111

24. Miao Y, Yang J, Alzahrani B, Lv G, Alafif T, Barnawi A, Chen M: Abnormal behavior learning based on edge computing toward a crowd monitoring system. IEEE Network. 2022, 36:90-96. 10.1109/mnet.014.2000523

25. Khanam S, Sharif M, Raza M, Ishaq W, Fayyaz M, Kadry S: Anomaly recognition in surveillance based on feature optimizer using deep learning. PLoS One. 2025, 20:e0313692. 10.1371/journal.pone.0313692

26. Wu P, Zhou X, Pang G, Sun Y, Liu J, Wang P, Zhang Y: Open-vocabulary video anomaly detection [PREPRINT]. arXiv. 2023, 10.48550/arXiv.2311.07042

27. Akhtaruzzaman M, Shafie AA, Khan R: Representation of human gait trajectory through temporospatial image modelling. ARPN Journal of Engineering and Applied Sciences. 2016, 11:4105-4110.

28. Penn-Fudan Database for Pedestrian Detection and Segmentation. (2026). Accessed: March 20, 2026: https://www.cis.upenn.edu/~jshi/ped_html/.

29. Ferryman JM, Shahrokni A: PETS2009: dataset and challenge in 2009. 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Snowbird, UT, USA. 2009, 1-6. 10.1109/PETS-WINTER.2009.5399556

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8

30. Lu C, Shi J, Jia J: Abnormal event detection at 150 FPS in MATLAB. 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia. 2013, 2720-2727. 10.1109/ICCV.2013.338

31. Cohen J: A coefficient of agreement for nominal scales. Educational and Psychological Measurement. 1960, 20:37-46. 10.1177/001316446002000104

32. Landis JR, Koch GG: The measurement of observer agreement for categorical data. Biometrics. 1977, 33:159-174. 10.2307/2529310

33. McNemar Q: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika. 1947, 12:153-157. 10.1007/BF02295996

How to cite this article:
Mushad R, Akhtaruzzaman M (November 19, 2026) A Multi-Mode Deep Learning Framework for Suspicious Behavior Detection in Controlled Environments. Cureus J Comput Sci 3 : es44389-026-00381-8. DOI https://doi.org/10.7759/s44389-026-00381-8