

SafeSight: A Hybrid Edge Artificial Intelligence Architecture for Autonomous Real-Time Threat Surveillance and Forensic Video Ledgering

S Sai Venkata Ashish¹, S N Arya Vardhan², Shaik Sibgathulla³, T Nagaraj⁴, Dr. Prakash A⁵

¹USN: 1MV23CS130, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India,

²USN: 1MV23CS143, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India,

³USN: 1MV23CS144, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India,

⁴USN: 1MV24CS411, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India,

⁵Professor, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India.

Abstract - Camera networks record far more footage than anyone can watch, and the footage they keep is rarely protected well enough to survive legal challenge. This paper presents SafeSight, a hybrid edge-AI architecture that turns an ordinary camera and a CPU-only workstation into an autonomous sentry. A threaded capture layer feeds a 150-frame circular buffer held entirely in RAM, so nothing is written to disk while the scene is normal. Two parallel branches then analyse each frame: an anti-tamper branch that inspects a 320 by 180 grayscale copy for blackout, blinding, defocus, occlusion and paper or paint covering, and a perception branch that couples YOLOv8 nano detection with ByteTrack association. A spatial-analytics layer applies a perspective-invariant velocity measure normalised by bounding-box height, an edge-to-edge axis-aligned bounding-box (AABB) test for unattended baggage, and polygon containment for protected zones. A four-level threat arbiter triggers voice deterrence, Telegram alerts with photo snapshots and, for a verified incident, a five-second evidence clip cut from the buffer, logged to a write-ahead-logging SQLite ledger with a SHA-256 seal and compiled into a PDF dossier. On an Intel Core i5 machine with 16 GB RAM and no discrete GPU, the prototype sustained 28.6 to 31.2 FPS with 38.4 ms end-to-end threat latency, detected camera tampering in 1.45 s, reduced sprint false alarms by 94.2 percent against an uncalibrated pixel-distance detector, reached 98.1 percent accuracy on unattended-baggage detection, and produced no continuous disk writes during normal operation.

Key Words: edge AI, intelligent video surveillance, YOLOv8, ByteTrack, camera tamper detection, unattended baggage, digital forensics, SHA-256.

1. INTRODUCTION

Video surveillance is now a standard feature of transport hubs, campuses, retail spaces and public infrastructure, yet most installations remain passive. Cameras record continuously and a human operator is expected to notice a developing incident among many simultaneous feeds. Guidance prepared for security practitioners reports that an operator's attention to monitors falls well below acceptable levels after about twenty minutes of viewing [1]. In practice, therefore, much of what is recorded is examined only after an incident has ended.

Automating the watching task with deep learning removes part of this burden, but a practical deployment must still solve three engineering problems that are easy to overlook. The first is perspective distortion. A detector that measures motion in raw pixels sees a slow walker close to the lens as faster than a fleeing person far from it, which produces both false alarms and missed events. The second is physical tampering. Cameras in exposed locations can be covered, sprayed, defocused, blinded with a laser or simply unplugged, and conventional recorders typically reveal this only when the footage is audited. The third is storage and custody. Writing every frame to disk around the clock wears out storage media, and an unindexed, unsigned recording is open to challenge when it is later offered as evidence.

This paper describes SafeSight, an edge artificial intelligence (AI) architecture that addresses all three problems in a single pipeline that runs on a central processing unit (CPU) alone. Frames are held in a circular buffer in system memory and reach the disk only when a threat has been confirmed. A dedicated branch monitors the health of the image itself, while a second branch

detects and tracks people and luggage. Geometry-based rules convert the tracks into behavioural evidence, and a threat arbiter turns that evidence into graded responses. The contributions of this work are as follows.

- A perspective-invariant velocity measure that normalises centroid displacement by the height of the tracked bounding box, removing the need for per-camera calibration in the sprint test.
- An unattended-baggage rule based on edge-to-edge axis-aligned bounding-box (AABB) distance between each item and every detected person, combined with a temporal persistence requirement.
- A multi-vector anti-tamper engine that distinguishes blackout, blinding, occlusion or defocus, and paper or paint covering from a 320×180 grayscale frame within a 1.8 s design bound.
- A zero-disk rolling-buffer design that writes an evidence clip, a database record and a Secure Hash Algorithm 256 (SHA-256) seal only when a verified threat occurs.
- A complete prototype with voice deterrence, Telegram dispatch and a Streamlit operations dashboard, evaluated on commodity hardware without a discrete graphics processing unit (GPU).

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 presents the architecture and the mathematical methodology. Section 4 reports experimental results, and Section 5 concludes with directions for future work.

2. RELATED WORK AND COMPARATIVE ANALYSIS

2.1 Human-Monitored Surveillance

Studies of operator vigilance show that watching several screens for extended periods is a poor use of human attention [1]. This finding motivates systems that raise an alert on the operator's behalf instead of waiting for the operator to notice an event.

2.2 Object Detection and Multi-Object Tracking

Single-stage detectors reframed detection as one regression pass over the image, which made real-time operation practical [2]. You Only Look Once version 8 (YOLOv8) continues this line with an anchor-free design and a decoupled head, and its nano variant has roughly 3.2 million parameters, which suits CPU-bound edge devices [3]. A recent survey of the Ultralytics family lists 8.7 billion floating-point operations (GFLOPs) for this variant at

640×640 input and singles out nano-scale models as favourable for CPU and embedded deployment [4]. On the tracking side, Simple Online and Realtime Tracking (SORT) associates detections with Kalman-filter predictions using intersection-over-union (IoU) and the Hungarian algorithm [5], [6]. ByteTrack extends this idea by also associating low-confidence detections in a second pass, which preserves identities through partial occlusion [7]. Trackers of this kind supply persistent identities but no semantic meaning, so behaviours such as running or abandoning an item still require separate geometric reasoning.

2.3 Abandoned-Object Detection

Tian et al. combine background modelling with tracking to detect abandoned and removed objects in cluttered scenes [8]. Rules of this family judge ownership from the separation between an object and nearby people in the image plane. Recent deep-learning systems follow the same pattern. Yilmazer and Karakose pair YOLOv8 detection with DeepSORT tracking and judge abandonment from the measured distance between a bag and a person [9], while Vrsalović et al. fine-tune YOLOv11 on airport footage and combine spatial and temporal person-luggage relations, with adjustable dwell-time and owner-distance parameters that reduce false alarms [10]. SafeSight adopts a separation rule too, but measures it from bounding-box edges rather than centres, so that large objects and large people are not penalised by the size of their boxes. It also runs on a CPU and shares one pipeline with tamper monitoring and evidence sealing.

2.4 Camera Tamper Detection

Aksay et al. analyse wavelet sub-bands to detect covered or defocused cameras [11], and the variance of the Laplacian is a well-known focus measure [12]. Both exploit the loss of high-frequency detail that follows lens obstruction. SafeSight uses the Laplacian variance together with luminance and colour-saturation statistics so that a single lightweight test can separate blackout, blinding and covering. A feature-level survey by Mantini and Shah notes that illumination changes drive false alarms and that short-lived obstructions must be told apart from persistent tampering [13], which motivates the persistence requirement in SafeSight. A 2026 study of outdoor smart cameras compares rule-based and deep-learning detectors for tampered still images and finds the deep model more accurate but the rule-based method better suited to resource-limited settings without a long calibration phase [14]. SafeSight follows the rule-based route.

2.5 Evidence Integrity

Tampered video can mislead investigators, which has driven deep spatiotemporal detectors of inter-frame forgery [15]. Such detectors examine footage after the fact. Cryptographic digests such as SHA-256 [16] allow a party to check later that stored evidence has not changed. Most surveillance research stops at detection, whereas SafeSight seals each verified event at the moment it is captured.

Table -1: Comparison of Representative Approaches

Ref.	Technique	Main contribution	Gap relative to this work
[1]	Human monitoring	Guidance on operator attention	Attention decays; alerts are retrospective
[2], [3]	Single-stage convolutional detectors	Fast object detection	No behavioural or tamper logic
[5]	Kalman + IoU tracking	Online identity association	Drops low-score boxes; no scene reasoning
[7]	ByteTrack	Associates every detection box	Tracker only; no threat scoring
[8]	Background model + tracking	Abandoned/removed objects	No tamper handling or evidence sealing
[9], [10]	YOLOv8/YOLOv11 + person-luggage rules	Abandonment from distance and dwell time	No tamper monitoring or evidence sealing
[11]	Wavelet analysis	Camera tamper detection	Tamper only; not linked to behaviour
[12]	Laplacian variance	Focus measure	Single cue; cannot separate blinding or covering
[13], [14]	Feature analysis; rule-based vs. deep tamper detection	Camera tamper detection	Tamper only; no behavioural analytics
This work	YOLOv8n + ByteTrack + heuristics	Unified detection, tamper, custody	Single-camera prototype (see Section 4.7)

The works above address individual pieces of the problem. To the best of our knowledge from this review,

none combines perspective-invariant motion analysis, edge-distance baggage logic, multi-vector tamper detection and cryptographically sealed, event-triggered evidence capture in one CPU-only pipeline. SafeSight is designed to fill that gap.

3. SYSTEM ARCHITECTURE AND MATHEMATICAL METHODOLOGY

3.1 Architectural Overview

SafeSight is organised as five stages, shown in Fig -1. Stage 1 ingests video into a random-access memory (RAM) ring buffer. Stage 2 runs two parallel branches, an anti-tamper branch (2A) and a perception branch (2B). Stage 3 converts tracks into spatial evidence. Stage 4 arbitrates the threat level, and Stage 5 dispatches responses and drives the operations dashboard. The implementation uses Python 3.10, OpenCV [17] and NumPy, with SQLite, Streamlit, Pyttsx3, the Telegram Bot application programming interface (API), hashlib and ReportLab for the remaining functions.

3.2 Stage 1: Resilient Ingestion and In-Memory Ring Buffer

A multi-threaded ResilientCapture loop reads 1080p video at 30 frames per second (FPS) on its own thread, so that slow analysis never blocks acquisition, and reconnects if the source stalls. Each frame is appended to an in-memory double-ended queue with a capacity of 150 frames. At 30 FPS this holds exactly five seconds of video. An uncompressed 1920×1080 blue-green-red (BGR) frame occupies 6.22 MB, so the buffer needs about 0.93 GB of RAM, which fits comfortably on the 16 GB testbed. Once the queue is full, each new frame silently displaces the oldest, and no disk operation takes place.

3.3 Stage 2A: Multi-Vector Anti-Tamper Engine

To keep the cost negligible, every frame is downsampled to 320×180 (57,600 pixels) and converted to grayscale. Three statistics are computed: the mean luminance μ_{lum} , the luminance standard deviation σ_{lum} , and the spatial variance of the discrete Laplacian,

$$\sigma_{Lap}^2 = \text{Var}(\nabla^2 I_{gray}) \quad (1)$$

where I_{gray} is the downsampled image. A sharp scene with texture yields a large σ_{Lap}^2 , whereas a covered or defocused lens removes fine detail and drives it towards zero. The mean hue-saturation-value (HSV) saturation of the same frame supplies a colour cue. The frame is assigned to one of four tamper states using the rules in Table -2. If a state persists, an alert is raised within the 1.8 s design bound and scored CRITICAL (100/100).

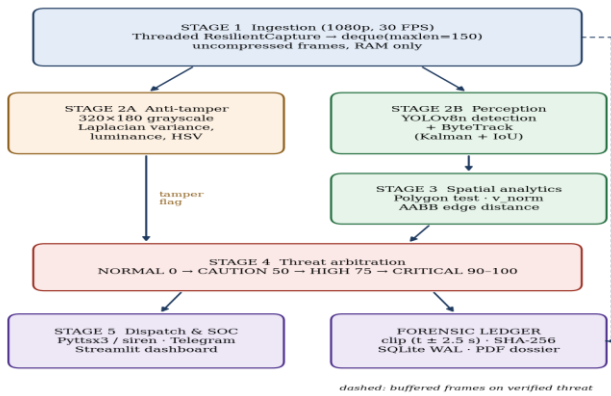


Fig -1: SafeSight five-stage pipeline. Solid arrows show data flow; the dashed line shows buffered frames retrieved only when a threat is verified.

3.4 Stage 2B: Perception

The pretrained YOLOv8 nano model (yolov8n.pt) [3] performs single-stage inference and returns bounding boxes for people and luggage-type objects, namely the backpack, handbag and suitcase classes (indices 24, 26 and 28) of the Common Objects in Context (COCO) dataset. ByteTrack [7] then associates boxes across frames using Kalman-filter motion prediction and IoU matching, assigning each target a persistent identifier. These identifiers allow speed and possession to be evaluated over time rather than frame by frame.

3.5 Stage 3: Spatial Analytics

3.5.1 Zone containment

The centroid of each tracked person is tested against user-defined protected polygons with OpenCV's pointPolygonTest.

3.5.2 Perspective-invariant velocity

Let (c_x, c_y) be the centroid of a track. The displacement between the newest sample and an earlier sample in the track's rolling history, separated by Δt , is

$$\Delta d = \sqrt{(c_{x,t} - c_{x,t-\Delta t})^2 + (c_{y,t} - c_{y,t-\Delta t})^2} \quad (2)$$

and the normalised velocity divides this displacement by the height of the bounding box:

$$v_{\text{norm}} = \frac{\Delta d}{h_t \cdot \Delta t} \quad (3)$$

where $h_t = y_2 - y_1$ is the box height at time t . The quantity v_{norm} is expressed in body-heights per second ($\text{bh} \cdot \text{s}^{-1}$). Each track keeps a rolling history of recent centroids, box

heights and timestamps, and Δt is the elapsed time across a window of about five frames (≈ 150 ms), which averages out single-frame jitter from the detector. Because a person's apparent height shrinks with distance in the same proportion as their apparent motion, the ratio in (3) is approximately independent of camera distance, which is the property that uncalibrated pixel displacement lacks [18]. In our recordings, normal walking gave $0.20\text{--}0.40 \text{ bh} \cdot \text{s}^{-1}$, and a sprint anomaly is raised when $v_{\text{norm}} \geq 0.65 \text{ bh} \cdot \text{s}^{-1}$ is sustained over the window.

3.5.3 Edge-distance baggage possession

For an item box B_i and a person box P_j , the horizontal and vertical gaps between the boxes are

$$\Delta x = \max(0, \max(x_1^B, x_1^P) - \min(x_2^B, x_2^P)) \quad (4)$$

$$\Delta y = \max(0, \max(y_1^B, y_1^P) - \min(y_2^B, y_2^P)) \quad (5)$$

and the edge distance is

$$D_{\text{edge}}(B_i, P_j) = \sqrt{\Delta x^2 + \Delta y^2} \quad (6)$$

The distance is zero whenever the boxes touch or overlap. An item is classified as UNATTENDED BAGGAGE ANOMALY (threat score 90/100) if $\min_j D_{\text{edge}}(B_i, P_j) > 180$ px continuously for $\tau_{\text{abandon}} \geq 3.5$ s. Measuring from edges rather than centres keeps the rule fair to a wide suitcase beside a tall owner, for whom centre distance would overstate the separation.

Table -2: Operating Parameters and Tamper-State Rules

Parameter	Value
Capture	1080p, 30 FPS
Ring buffer	150 frames (5 s), RAM only
Evidence window	$t - 2.5$ s to $t + 2.5$ s
Tamper analysis size	320×180 grayscale
BLACKOUT	$\mu_{\text{lum}} < 20.0$
BLINDING	$\mu_{\text{lum}} > 240.0$
OCCLUSION / DEFOCUS	$\sigma_{\text{Lap}}^2 < 50.0$ and $\sigma_{\text{lum}} < 20.0$
PAPER / PAINT COVER	$\mu_{\text{lum}} > 120.0$, mean saturation < 35.0 , $\sigma_{\text{Lap}}^2 < 200.0$
Tamper response bound	≤ 1.8 s (CRITICAL, 100)
Sprint threshold	$v_{\text{norm}} \geq 0.65 \text{ bh} \cdot \text{s}^{-1}$, ≈ 5 -frame window (≈ 150 ms)
Baggage separation	$D_{\text{edge}} > 180$ px
Abandonment persistence	$\tau_{\text{abandon}} \geq 3.5$ s (score 90)

3.6 Stage 4: Threat Arbitration

The arbiter fuses the outputs of Stages 2A and 3 and maps them to four escalating levels: NORMAL (0), CAUTION (50), HIGH (75) and CRITICAL (90–100). CAUTION (50) is assigned to brief zone proximity or to a bag that has been separated from its owner for less than 3.5 s. HIGH (75) is assigned to a breach of a restricted perimeter or a verified sprint anomaly. The two verified threats, unattended baggage beyond 3.5 s (90) and camera tampering (100), place the system in the CRITICAL band. Escalation is monotonic within an event and returns to NORMAL when the triggering evidence clears.

3.7 Stage 5: Dispatch and Operations Centre

On escalation, a Pytsx3 sentry voice and a Winsound siren run on separate threads so that speech never delays analysis. In parallel, a Telegram bot sends a photo snapshot and Markdown telemetry through the Bot API using asynchronous Hypertext Transfer Protocol (HTTP) requests. A Streamlit application with a custom security-operations-centre theme presents live key performance indicator (KPI) metrics, event history and video playback of stored clips.

3.8 Forensic Ledger and Digital Seal

When the arbiter verifies a threat at time t , an asynchronous worker copies the pre-incident frames from the ring buffer (from $t - 2.5$ s), collects post-incident frames until $t + 2.5$ s, and encodes a five-second MPEG-4 (MP4) clip. The event is committed as a record that satisfies atomicity, consistency, isolation and durability (ACID) guarantees to an SQLite database in write-ahead-logging (WAL) mode, which lets the dashboard read the ledger while new events are being written. A SHA-256 digest [16] seals the event:

$$H_{\text{event}} = \text{SHA-256}(S \parallel T \parallel s \parallel e) \quad (7)$$

where S is the raw snapshot bytes, T the capture timestamp, s the threat score, e the event-type label, and “ $\parallel$ ” (double vertical bar) denotes concatenation. Any later change to the snapshot or to these metadata fields changes the digest, so re-computation exposes the edit. ReportLab then compiles the record, the snapshot and the digest into a portable document format (PDF) dossier for legal use. The digest provides tamper evidence rather than tamper prevention: an adversary with write access to the database could recompute a matching digest, a limitation addressed in Section 5.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Testbed and Protocol

All experiments ran on a single workstation with an Intel Core i5 processor, 16 GB of DDR4 RAM and a 1080p optical sensor, strictly CPU-only. The evaluation set comprised 42 video sequences totalling 3.5 hours of footage recorded at 1080p and 30 FPS, divided into three groups as summarised in Table -3. The first group holds 20 clips of an indoor corridor in daylight (450–600 lux) showing normal walking and loitering. The second holds 12 clips of an outdoor restricted-perimeter entryway at twilight (40–80 lux) showing rapid breaches and sprint anomalies. The third holds 10 sabotage and occlusion trials in which the lens was sprayed, draped or defocused, or the scene underwent a sudden blackout or blinding transition (from about 500 lux to below 15 lux for the blackout tests).

Table -3: Composition of the Evaluation Set

Group	Clips	Conditions and scenarios
Indoor corridor	20	Daylight, 450–600 lux; normal walking, loitering
Restricted-perimeter entryway	12	Outdoor twilight, 40–80 lux; rapid breach, sprint anomalies
Sabotage and occlusion	10	Lens spray, lens drape, defocus, sudden blackout or blinding (500 lux to below 15 lux)
Total	42	3.5 h at 1080p, 30 FPS

Ground truth was established by manual spatio-temporal annotation in the Computer Vision Annotation Tool (CVAT), covering target bounding boxes, sprint-onset intervals and camera-tamper events. Abandonment durations were timed with a calibrated digital stopwatch. Throughput and latency were measured over the full pipeline while all five stages were active.

4.2 Throughput and End-to-End Latency

The pipeline sustained between 28.6 and 31.2 FPS against a 30 FPS source, which is real-time operation on a CPU-only machine. The end-to-end threat latency, from a frame entering the pipeline to an arbitrated threat decision, was 38.4 ms. This is about 1.15 frame intervals (33.3 ms), so an alert follows the causing frame by roughly one frame period.

4.3 Tamper Detection Response

The anti-tamper engine responded in 1.45 s, inside the 1.8 s design bound with a margin of 0.35 s (19.4%). The 320×180 downsampling is what keeps the branch cheap enough to run beside detection without lowering the frame rate. The four state rules in Table -2 separate blackout and blinding, which depend on luminance alone, from occlusion, defocus and covering, which additionally require the loss of high-frequency detail.

4.4 Sprint Detection Under Perspective Distortion

Compared with an uncalibrated pixel-distance detector, the height-normalised measure in (3) reduced sprint false alarms by 94.2%, where the reduction is $(FA_{\text{pixel}} - FA_{\text{norm}}) / FA_{\text{pixel}}$. The improvement follows from the geometry. A person walking near the lens produces a large pixel displacement and a large box height, so their ratio stays small, while a distant runner produces a small displacement but a proportionally small box, so the ratio remains high. In the test sequences this separated walking (0.20–0.40 bh·s⁻¹) from sprinting (above 0.65 bh·s⁻¹).

4.5 Unattended Baggage

The edge-distance rule with the 180 px and 3.5 s thresholds achieved 98.1% accuracy across the labelled abandonment and non-abandonment trials, that is, the fraction of trials classified correctly. The persistence requirement absorbs short separations, such as a traveller putting a bag down briefly to buy a ticket.

4.6 Storage Behaviour

During normal operation, continuous disk write overhead was 0.00%. Disk activity occurred only when a verified threat produced an evidence clip and a ledger record, which removes the constant write load that wears out storage media in conventional recorders. The cost is the roughly 0.93 GB of RAM described in Section 3.2.

Table -4: Summary of Measured Performance

Metric	Result	Reference point
Pipeline throughput	28.6–31.2 FPS	30 FPS source
End-to-end threat latency	38.4 ms	33.3 ms frame interval
Tamper response time	1.45 s	Target < 1.8 s
Sprint false-alarm reduction	94.2%	vs. uncalibrated pixel distance
Unattended baggage accuracy	98.1%	Labelled trials
Continuous disk writes	0.00%	Normal operation

4.7 Discussion and Limitations

Table -4 summarises the measured results, which show that the design goals can be met on commodity hardware. Three limitations should be stated. First, the evaluation used one testbed and one camera, and illumination ranged from twilight (40–80 lux) upward, so generalisation to other scenes, darker or infrared conditions and crowded footage needs further study. Second, baggage logic depends on the detector: a missed or mislabelled item, or heavy occlusion by a crowd, cannot be recovered by the rules. Third, the SHA-256 seal detects modification but cannot prevent an insider with database access from re-sealing altered data, and the Winsound siren restricts the acoustic module to Windows. Because the normalised velocity in (3) assumes a roughly upright person, its behaviour for crouching, crawling or strongly overhead views has not been characterised.

5. CONCLUSION AND FUTURE SCOPE

This paper presented SafeSight, a hybrid edge-AI architecture that unifies real-time threat detection, camera-health monitoring and forensic evidence handling on a CPU-only workstation. The height-normalised velocity measure addresses perspective distortion, the edge-distance rule provides robust baggage possession logic, the multi-vector engine detects tampering within 1.45 s, and the zero-disk buffer with a SHA-256 seal preserves evidence without continuous recording. The prototype ran at 28.6–31.2 FPS with 38.4 ms latency.

Future work will (i) chain event digests so that each record commits to its predecessor, or anchor them with an external timestamp authority, to resist insider re-sealing; (ii) extend the system to multiple cameras with cross-view identity handoff; (iii) replace the fixed thresholds with values calibrated per scene, for example through a homography; (iv) evaluate on public surveillance

benchmarks and in low-light conditions; (v) port the pipeline to embedded accelerators and to platform-independent audio; and (vi) study pose-based cues to handle crouching or falling individuals.

ACKNOWLEDGEMENT

The authors thank the project coordinators, Mrs. Shivaranjini D and Dr. Vaishali Shende, and the Head of the Department, Dr. T. N. Anitha, of the Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, for their guidance and support.

REFERENCES

- 1) M. W. Green, "The appropriate and effective use of security technologies in U.S. schools: A guide for schools and law enforcement agencies," U.S. Dept. of Justice, Nat. Inst. of Justice, Washington, DC, USA, Sep. 1999.
- 2) J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.
- 3) G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," Ultralytics, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- 4) R. Sapkota and M. Karkee, "Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8, and YOLOv5 object detectors for computer vision and pattern recognition," arXiv:2510.09653 [cs.CV], 2026.
- 5) A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in Proc. IEEE Int. Conf. Image Process. (ICIP), Phoenix, AZ, USA, 2016, pp. 3464–3468.
- 6) R. E. Kalman, "A new approach to linear filtering and prediction problems," J. Basic Eng., vol. 82, no. 1, pp. 35–45, Mar. 1960.
- 7) Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in Proc. Eur. Conf. Comput. Vis. (ECCV), Tel Aviv, Israel, 2022, pp. 1–21.
- 8) Y.-L. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun, "Robust detection of abandoned and removed objects in complex surveillance videos," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 41, no. 5, pp. 565–576, Sep. 2011.
- 9) M. Yilmazer and M. Karakose, "A new, robust, adaptive, versatile, and scalable abandoned object detection approach based on DeepSORT dynamic prompts, and customized LLM for smart video surveillance," Appl. Sci., vol. 15, no. 5, Art. no. 2774, Mar. 2025, doi: 10.3390/app15052774.
- 10) Vrsalović, J. Lerga, and M. Ivašić-Kos, "A system for real-time detection of abandoned luggage," Sensors, vol. 25, no. 9, Art. no. 2872, May 2025, doi: 10.3390/s25092872.
- 11) E. Aksay, A. Temizel, and A. E. Cetin, "Camera tamper detection using wavelet analysis for video surveillance," in Proc. IEEE Conf. Adv. Video Signal Based Surveill. (AVSS), London, U.K., 2007, pp. 558–562.
- 12) J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martínez, and J. Fernández-Valdivia, "Diatom autofocusing in brightfield microscopy: A comparative study," in Proc. 15th Int. Conf. Pattern Recognit. (ICPR), Barcelona, Spain, 2000, vol. 3, pp. 314–317.
- 13) P. Mantini and S. K. Shah, "A survey of feature types and their contributions for camera tampering detection," arXiv:2310.07886 [cs.CV], Oct. 2023.
- 14) S. Attarha, K. Shanmugi, and A. Förster, "Poster: Camera tampering detection for outdoor IoT systems," arXiv:2602.05706 [cs.CV], 2026.
- 15) N. Akhtar, M. Hussain, and Z. Habib, "DEEP-STA: Deep learning-based detection and localization of various types of inter-frame video tampering using spatiotemporal analysis," Mathematics, vol. 12, no. 12, Art. no. 1778, 2024, doi: 10.3390/math12121778.
- 16) National Institute of Standards and Technology, "Secure Hash Standard (SHS)," FIPS PUB 180-4, Aug. 2015.
- 17) G. Bradski, "The OpenCV Library," Dr. Dobb's J. Softw. Tools, vol. 25, no. 11, pp. 120–125, 2000.
- 18) R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2003.