

Article

Not peer-reviewed version

Prototype of a Smart Security System Using Facial Recognition and Body Language Analysis in Residential Complexes

Armando García Saavedra , Mauricio Dassel Corona Reyes , Jorge David Orozco Guillén ,
[Idalia Maldonado Castillo](#) , Rodolfo Romero Herrera *

Posted Date: 14 October 2026

Keywords: facial recognition; pose estimation; weapon detection; pgvector; WebSockets; computer vision; smart security; real-time incidents; artificial intelligence; data science



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Prototype of a Smart Security System Using Facial Recognition and Body Language Analysis in Residential Complexes

Armando García Saavedra, Mauricio Dassel Corona Reyes, Jorge David Orozco Guillén, Idalia Maldonado Castillo and Rodolfo Romero Herrera *

Escuela Superior de Cómputo, Instituto Politécnico Nacional, Mexico City, Mexico

* Correspondence: rromero@ipn.mx

Abstract

This paper presents a prototype of a smart security system for residential complexes that integrates real-time facial recognition, aggressive body language analysis, and weapon detection. Its service architecture utilizes a backend featuring hybrid persistence (PostgreSQL, pgvector, and MinIO) alongside WebSockets for immediate alert dissemination. A centralized rules engine correlates detections from individual modules (faces, poses, and YOLO models) into actionable incidents, thereby optimizing response capability and residential surveillance.

Keywords: facial recognition; pose estimation; weapon detection; pgvector; WebSockets; computer vision; smart security; real-time incidents; artificial intelligence; data science

1. Introduction

1.1. Context and Problem Statement

Urban insecurity is one of the most persistent concerns of Mexican society. According to the National Urban Public Safety Survey (ENSU) conducted by INEGI in September 2025, 63% of the population aged 18 and older residing in the country's 91 urban areas considered their city unsafe [1]. Within the specific context of residential complexes, this challenge translates into burglaries, unauthorized intrusions, and criminal activity that traditional surveillance schemes—manually supervised closed-circuit television (CCTV) and access-control booths—fail to mitigate consistently. Reliance on the human factor introduces cognitive fatigue, missed suspects, delayed responses, and high operating costs [2].

1.2. Technological Gap Between Public and Private Video Surveillance

In the public sector, command centers such as Mexico City's C5 already employ artificial intelligence for real-time behavioral pattern analysis [3], and companies such as Microsoft offer computer vision for enterprise biometric authentication [4]. However, these solutions have not been effectively transferred to the private residential environment for three reasons:

- Scale and public focus: systems such as C5 are designed for large-scale urban surveillance, unsuited to the budget or operation of a residential complex.
- Closed-Set paradigm: conventional biometric classifiers assume a fixed number of identities, whereas a residential environment requires recognizing unenrolled visitors and service providers without retraining the model—that is, operating in an Open-Set modality.
- Single-criterion systems: private commercial solutions typically offer isolated modules (license-plate readers or facial verifiers) without an engine that correlates, in time and space, the appearance of an unknown face with an aggression pattern or the carrying of a weapon.

To address this gap, this work set out to develop a prototype smart security system for residential complexes that combines facial recognition, posture analysis, and object detection in order to identify authorized and unauthorized access, physical aggression, and weapon possession in real time; this included implementing facial recognition for residents and visitors, a posture-analysis model to detect aggression, a detector for firearms and edged weapons, a web interface for notifications and incidents, and a centralized database for historical security analysis.

2. Theoretical Framework and State of the Art

2.1. Facial Recognition: From Handcrafted Features to Deep Metric Learning

Before the rise of deep learning, techniques such as Eigenfaces (based on Principal Component Analysis), Fisherfaces, and Local Binary Patterns extracted hand-engineered facial features, but proved fragile under changes in illumination, scale, and pose. Convolutional neural networks made it possible to reformulate recognition as a classification problem via Softmax over a fixed set of identities; however, this approach inherits two structural limitations for security applications: the closed-set problem—an intruder is necessarily classified as the most similar resident—and the lack of scalability, since adding a new user requires retraining the entire network.

Deep Metric Learning resolves both limitations by learning a function that maps each face to a continuous vector (embedding), such that samples from the same person form compact clusters while those from different people remain separated; enrolling a new resident is thus reduced to computing and storing their reference vector. FaceNet popularized the Triplet Loss function, which requires the distance between an anchor and a positive sample to be smaller—by a margin m —than the distance to a negative sample, although it requires costly triplet mining during training.

ArcFace (Additive Angular Margin Loss) [5] overcame this limitation by introducing an additive angular margin directly in the angular space of a normalized hypersphere. By fixing the bias at zero and normalizing both the weights and the feature vectors, the classifier's logits reduce to the cosine of the angle between the input vector and each class center, scaled by a fixed factor s . Adding an angular margin m to the true class's angle forces the model to compress same-identity samples into tighter angular clusters and to geodesically separate the centers of different classes—a property that is key to distinguishing residents from intruders via a simple cosine-distance threshold.

In the prototype's persistence layer, comparison between embeddings is not performed by an additional neural network, but through a purely geometric operation: since the vectors are L2-normalized, the dot product between two embeddings equals the cosine of the angle separating them, and the cosine distance is obtained by subtracting that value from one. This property is what makes it possible to delegate the search for the most similar resident to a vector database engine—in this case, PostgreSQL's pgvector extension—rather than manually scanning every record with the neural network model on each verification.

2.2. Human Pose Estimation and Spatiotemporal Graphs

Pose estimation represents the human body as a set of interconnected joints, which provides robustness against changes in texture, clothing, or lighting. OpenPose [6] introduced Part Affinity Fields to reconstruct multi-person skeletons in real time, and RMPE [7] improved stability under fast movement and low-resolution cameras. To interpret the temporal evolution of movement—necessary to distinguish a shove from a mere gesture—graph convolutional networks (GCNs) replace dense video processing. The ST-GCN model [8] represents the body sequence as a directed spatio-temporal graph, with spatial edges connecting joints within the same frame and temporal edges linking the same node across consecutive frames.

2.3. Object Detection and Single-Stage Architectures

Early object detectors (Viola-Jones, HOG, DPM) used sliding windows over handcrafted descriptors. With the advent of CNNs came two-stage detectors (R-CNN, Fast R-CNN, Faster R-CNN), accurate but computationally expensive. To operate in real time, single-stage detectors—YOLO [9], SSD [10], and RetinaNet [11]—reformulated detection as a single regression problem over a grid of cells, where each cell predicts bounding boxes together with a confidence score and class probabilities. YOLOv10, adopted in this work, follows this paradigm and is complemented by the ByteTrack tracking algorithm to associate detections over time.

In the specific domain of weapon detection, the main challenges are not about average precision but about robustness against visually similar objects: a remote control, a tool, or a phone held in the hand frequently generates false positives in models trained only on positive images. For this reason, much of the recent literature in this domain focuses less on proposing new detection architectures and more on data-curation strategies—negative sampling, hard-negative mining—and on temporal-confirmation mechanisms that exploit video continuity, rather than deciding from a single isolated frame.

2.4. Comparison with the State of the Art

Table 1 compares the most representative solutions in the literature with the proposal put forward in this work.

Table 1. Comparison between smart video-surveillance systems and the proposed solution.

System	Scope and characteristics	Gap relative to the proposed solution
DRISHTI [12]	Urban video surveillance: analyzes age, face, gait, and possible weapons via distributed CCTV. Winner of the Smart India Hackathon 2020.	Designed for large-scale public/government surveillance; not optimized for private residential complexes.
Abba et al. framework [15]	Real-time object detection, tracking, and recognition via background subtraction, median filtering, and CNNs.	Focused on generic object tracking; lacks facial biometric verification and aggressive-behavior classification.
SmartFace (Innovatrics) [16]	Commercial real-time facial recognition platform compatible with IP cameras.	Highly specialized in faces; does not natively include behavior classification (hitting, fighting).
Proposed solution	Open-Set facial recognition, kinematic body analysis (ST-GCN), and weapon detection (YOLOv10) over an edge inference mesh.	Purpose-built for the private residential domain, with spatiotemporal correlation of events and no licensing costs.

3. System Architecture

3.1. Functional and Technical Scope

The system processes continuous video streams from network cameras (IP/RTSP) through three parallel artificial-intelligence modules—facial, body, and object—and transforms their detections into events, incidents, prioritized alerts, and queryable multimedia evidence. From a technical standpoint, it integrates a backend in FastAPI, relational and vector storage (PostgreSQL + pgvector), immutable object-based evidence storage (MinIO), and a reactive web interface (React) with notifications via WebSockets.

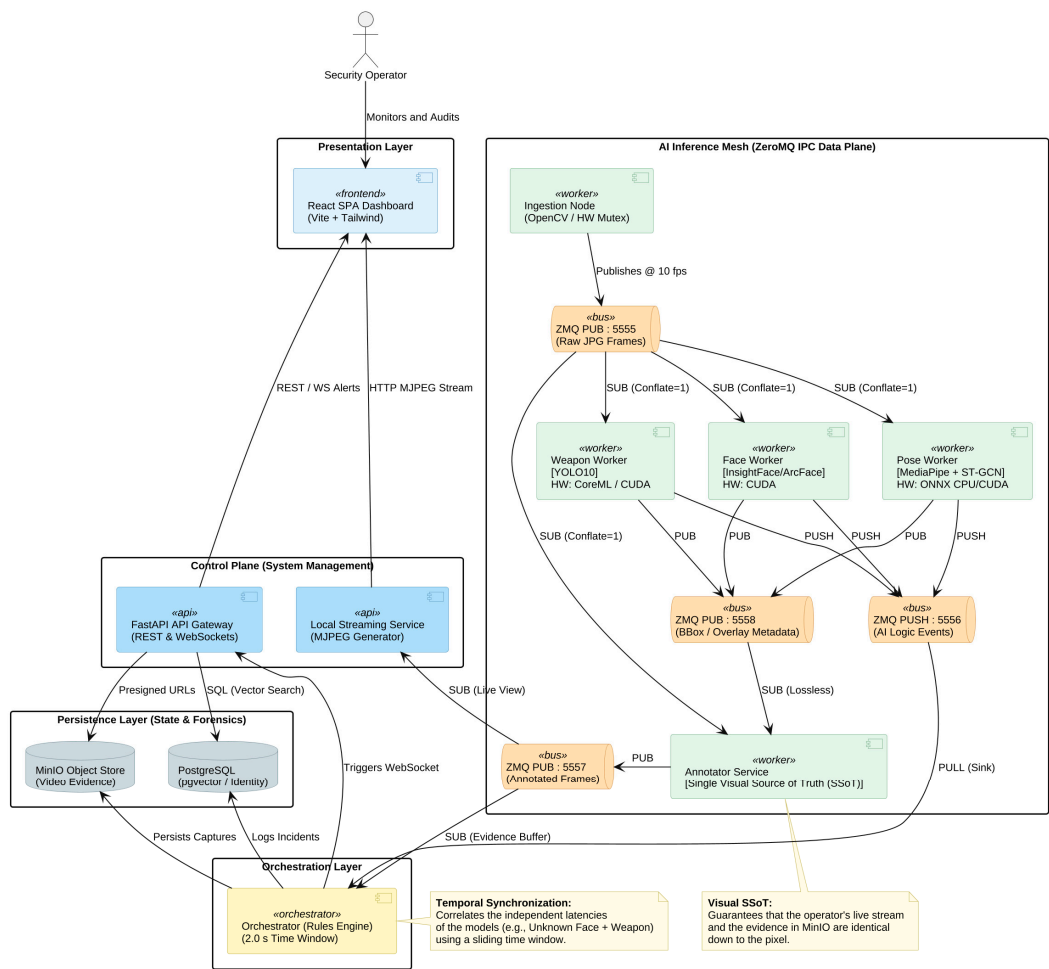


Figure 1. TT2 Residential Security System—enterprise architecture.

3.2. Actors and Use Cases

The operational domain is delimited by six actors: the resident (seeks entry with a minimal false-rejection rate), the unregistered person (represents the primary threat vector), the access camera, the vision/AI core, the alerts module, and the operator who audits the evidence. Table 2 summarizes the main use cases.

Table 2. Main use cases of the prototype.

Use case	Precondition	Expected outcome
Authorized resident entry	Registered resident; camera captures their face	Access granted and event logged with visual evidence
Unknown person in a restricted area	Facial match fails (score < threshold)	Alert sent. The event is logged for review.
Compound event (unknown + suspicious behavior)	Unidentified person and anomalous action occurring simultaneously	Immediate notification to security and event logged with visual evidence.
Repeated failed access	Consecutive failed attempts above the threshold	A warning alert is issued and the event is logged for further analysis.

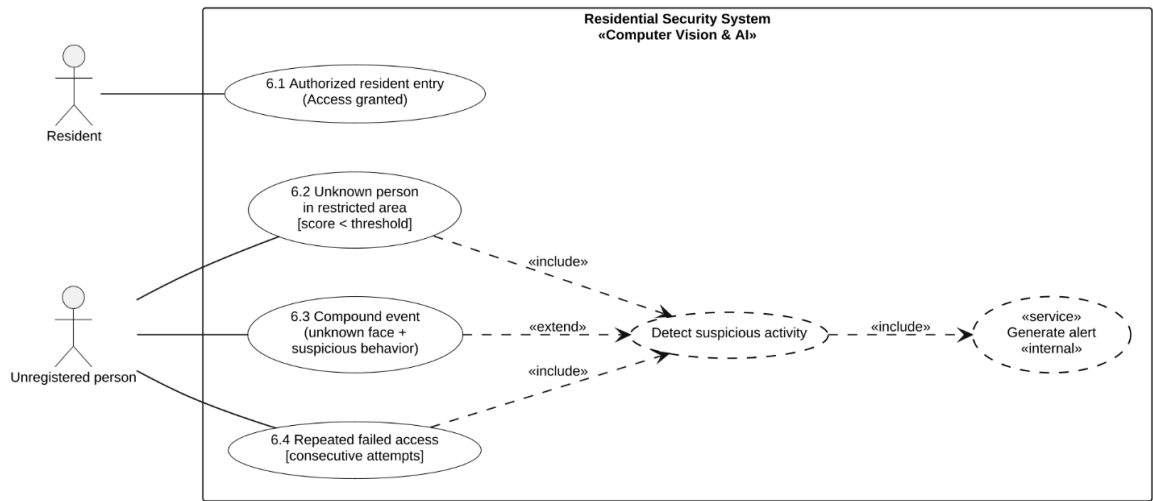


Figure 2. Use-case diagram.

3.3. Non-Functional Requirements and Business Rules

The system is governed by four key non-functional requirements: real-time performance with an effective throughput of 7 to 8 FPS and sub-second latency; biometric accuracy with a false-acceptance rate close to zero; identity scalability without retraining, thanks to ArcFace’s metric learning; and environmental robustness against changes in pose and lighting through affine facial alignment. The core business logic—which translates the networks’ probabilistic outputs into security decisions—is summarized in the threat-fusion rule, described further below.

3.4. Distributed Topology and Messaging Layer

To avoid the bottlenecks of a monolithic model, the system adopts an Event-Driven Distributed Architecture organized into three planes: the control plane (FastAPI and the React dashboard, which handle REST requests and WebSockets), the data-and-inference plane (an inter-process communication network based on ZeroMQ with no intermediaries), and the orchestration layer (the rules service, which applies a 2.0-second sliding time window and persists the forensic evidence).

The choice of ZeroMQ over traditional brokers (RabbitMQ, Kafka, Redis) responds to the need for microsecond-level latency in frame transfer. Table 3 summarizes the comparison.

Table 3. Brokerless messaging (ZeroMQ) versus a traditional broker.

Feature	ZeroMQ (chosen)	Traditional broker
Architecture	Decentralized (socket to socket)	Centralized (mandatory intermediary server)
Latency	Microseconds	Milliseconds
Memory overhead	Minimal (zero-copy, in-memory)	High (serialization + network hops)
Single point of failure	Distributed	Centralized (broker failure collapses the system)
Ideal use case	High-frequency IPC (video, sensors)	Transactional persistence of heavy events

3.4.1. Backpressure Mitigation with ZMQ_CONFLATE

To avoid the lag caused by cumulative FIFO queues, the AI workers’ sockets enable the ZMQ_CONFLATE=1 directive, which limits the network queue to exactly one message per subscriber. When the ingestion thread publishes a new frame while a model is still processing a previous one, the buffer atomically overwrites the obsolete intermediate frames, so that the next read always retrieves the most recent frame captured by the camera.

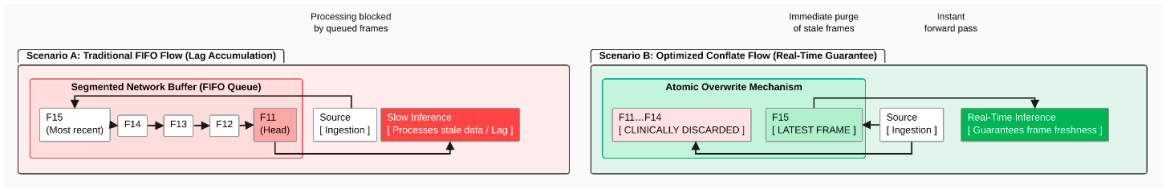


Figure 3. Backpressure-mitigation mechanics.

3.4.2. Causal Traceability and Contract Matrix

Each JSON message emitted includes an immutable header with the capture timestamp identifier, the camera identifier, and the MLflow calibration run identifier, which makes it possible to re-construct the complete timeline in the orchestrator. Table 4 summarizes the routing between components.

Table 4. Communication contract matrix between components.

Source	Destination	ZMQ Pattern	Payload
Video ingestion	Inference workers / annotator	PUB / SUB	Compressed JPEG frame (quality 75)
AI workers (facial / pose / objects)	Rules orchestrator	PUSH / PULL	Semantic logical events in JSON
AI workers (facial / pose / objects)	Annotator service	PUB / SUB	Geometric coordinates (boxes, key-points)
Annotator (single source of truth)	Orchestrator and web streaming	PUB / SUB	Annotated binary frame

3.5. Video Ingestion and Visual Consistency

A single ingestion thread takes exclusive control of the camera through the single-producer, multiple-consumer (SPMC) pattern, avoiding contention for simultaneous access to the physical device. Each captured frame is compressed to JPEG at quality 75, reducing its size from roughly 1.2 MB to a 35–40 KB profile (a bandwidth savings of over 95%), and is distributed at a sustained 10 FPS to the subscribed consumers.

To prevent divergence between what the operator observes and what is stored as forensic evidence, the system applies the Single Source of Visual Truth principle: the AI models do not draw on the image but instead emit only coordinates and probabilities; a centralized annotator service performs the single graphical-drawing pass, and the resulting frame is simultaneously forked toward the live interface and toward the evidence buffer, guaranteeing pixel parity between both.

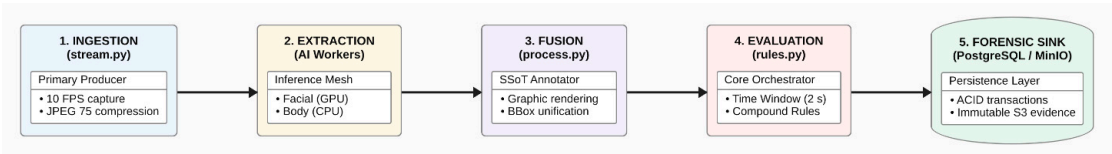


Figure 4. Sequential data lineage of the transactional pipeline (data life cycle).

3.6. Polyglot Persistence Layer

The data infrastructure separates structured information, biometric vectors, and immutable media files. PostgreSQL 16 hosts the persons table (includes the 512-dimensional biometric vector), cameras, incidents (with analytical metadata in JSONB columns), incident_timeline (links MinIO evidence to each incident), and alerts (state machine: unread, acknowledged, resolved). All primary keys are UUIDv4, which immunizes the system against enumeration attacks.

To avoid the so-called “double-write problem” — the risk of a resident’s metadata and their bio-metric vector becoming desynchronized when the two are stored in separate databases—, the facial signature is stored directly in PostgreSQL via the pgvector extension, with an HNSW (Hierarchical Navigable Small World) index optimized for cosine distance. At runtime, the ef_search parameter is set to 32, which keeps query latency consistently below 5 ms. In addition, the query returns only the name and the computed distance, discarding the 512-dimensional vector from the response and thereby reducing internal network traffic by more than 95%.

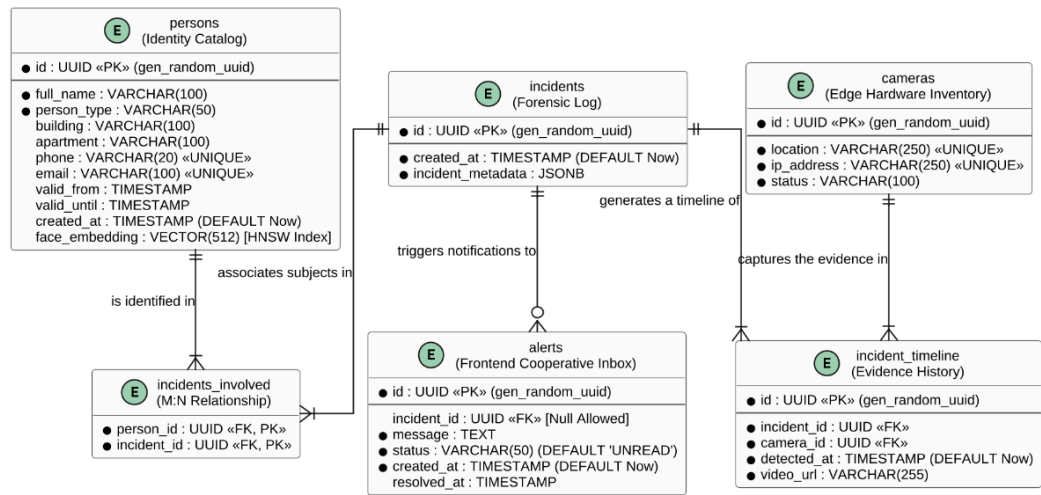


Figure 5. Entity–relationship diagram for the security system.

3.7. Operating Interface

The frontend was developed as a single-page application (SPA) in React, TypeScript, Vite, and Tailwind CSS. It is organized into seven modules: metrics dashboard, live monitoring, alert inbox, incident case files, resident administration with three-capture biometric enrollment, access-control simulation, and hot threshold configuration.

4. Artificial Intelligence Modules

4.1. Artifact Governance (MLOps)

The handling of large kinematic and visual datasets (such as NTU RGB+D [17]) is decoupled from the source code via Data Version Control (DVC): heavy binary files are stored in a remote object repository, while Git retains only lightweight pointers containing the data hash. Training is instrumented with MLflow, which generates a unique identifier capable of linking the code, the data, and the telemetry of each trained model, enabling updates or rollbacks without redeploying the backend.

To ensure reproducibility, the global seeds of the random-number generators are fixed and the cuDNN engine is forced to use only deterministic algorithms. The final models are exported to the open ONNX format, which enables two optimizations: constant folding (precomputing fixed matrix operations, such as the spatial graph’s adjacency matrix) and fusing the convolution, batch-normalization, and activation layers into a single operator.

4.2. Facial Recognition Module (ArcFace + Pgvector)

The facial module operates under the Open-Set paradigm and transforms the video stream into continuous 512-dimensional embeddings through a four-stage pipeline.

4.2.1. Preprocessing and Canonical Alignment

RetinaFace, a single-shot network coupled with a feature pyramid (FPN), detects the face’s bounding box and locates five landmark points. From these five points, an affine transformation is computed that rotates, translates, and scales the face to a standard resolution of 112 × 112 pixels, eliminating variations in posture or head tilt.

4.2.2. Feature Extraction and L2 Normalization

The aligned face is processed with a GPU-accelerated ResNet-50 network (InsightFace’s bufalo_1 package), which produces a dense 512-dimensional vector *v*. Before any comparison, the vector is normalized to L2 norm:

$$\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2} = \frac{\mathbf{v}}{\sqrt{\sum_{i=1}^{512} v_i^2}}$$

(1)

This operation constrains the biometric signature to the surface of a unit hypersphere, making the result independent of scene lighting and allowing similarity to be evaluated as a simple dot product.

4.2.3. Enrollment and Master Vector

Enrolling each resident requires three independent facial captures. Each image yields a unit vector *v*₁, *v*₂, *v*₃; their centroid is computed and re-normalized to L2 to obtain the master vector that is persisted in the database:

$$\mathbf{c} = \frac{1}{3} \sum_{k=1}^3 \mathbf{v}_k, \quad \mathbf{u}_{\text{master}} = \frac{\mathbf{c}}{\|\mathbf{c}\|_2}$$

(2)

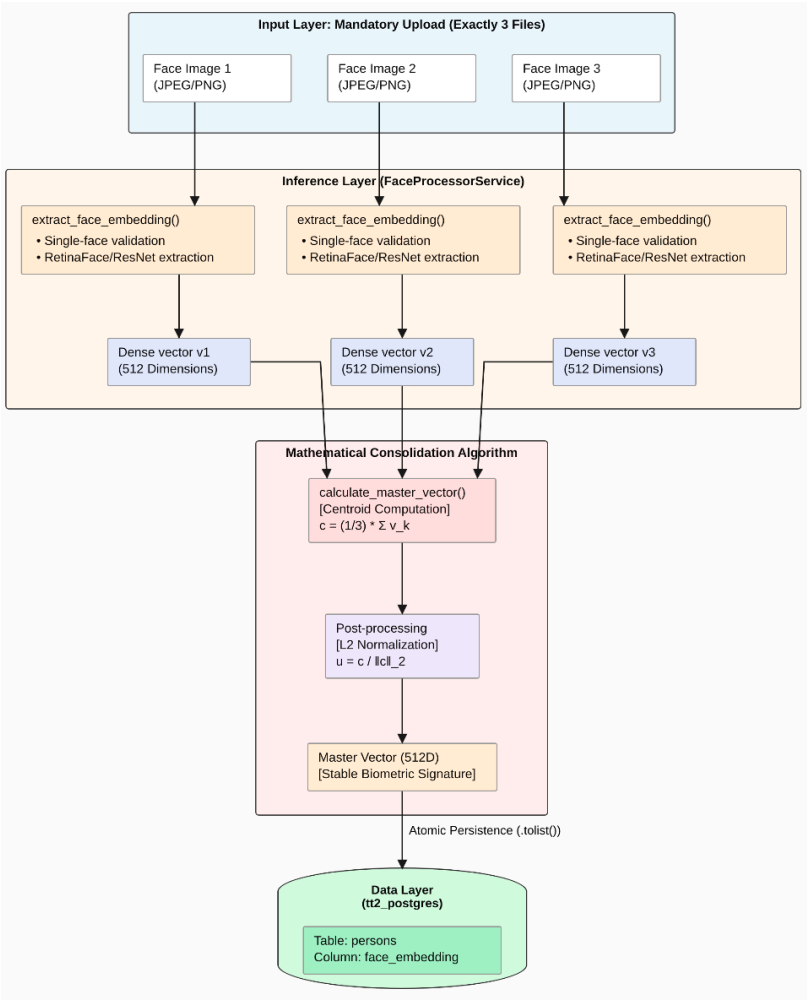


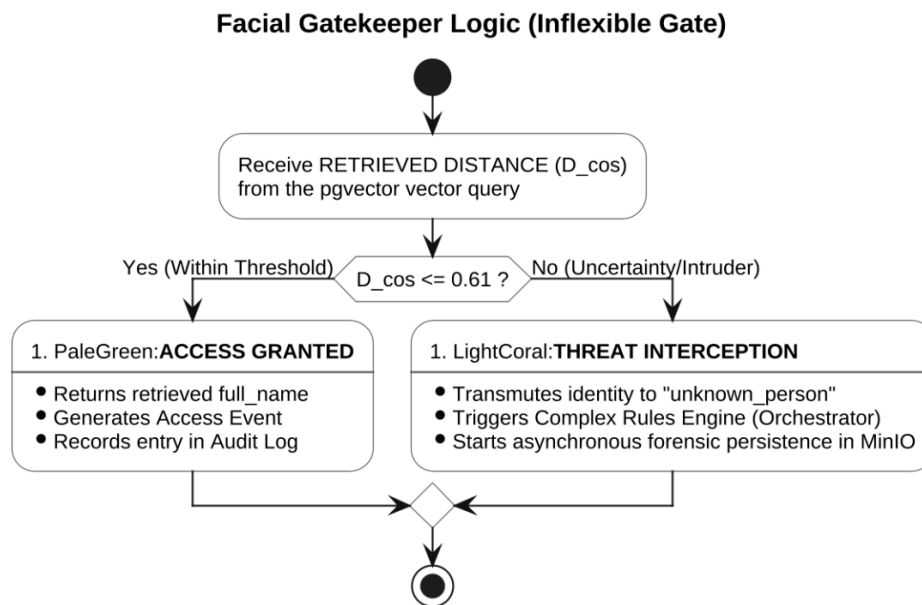
Figure 6. Biometric-enrollment pipeline and master-vector generation.

4.2.4. Gatekeeper Pattern and Threshold Calibration

The similarity between two unit vectors is evaluated using cosine distance:

$$D_{\cos}(\mathbf{u}, \mathbf{v}) = 1 - S_{\cos}(\mathbf{u}, \mathbf{v}) = 1 - (\mathbf{u} \cdot \mathbf{v}) \quad (3)$$

From a class-separability study on the Labeled Faces in the Wild dataset [18], it was determined that the similarity distributions of genuine residents and impostors intersect (Equal Error Rate) at a cosine similarity of 0.29, equivalent to $D_{\cos} = 0.71$. Operating at that point would imply an identical rate of false acceptances and false rejections, unacceptable in a residential environment. Therefore, the operating threshold was shifted toward the stricter region of the curve, setting the decision boundary at $D_{\cos} \leq 0.61$ (similarity ≥ 0.39), which reduces the false-acceptance rate to a value close to zero. When the distance retrieved from the HNSW index exceeds this threshold, the identity is forcibly transmuted to “unknown person” (Gatekeeper pattern).

**Figure 7.** Gatekeeper pattern and operating threshold.

4.3. Kinematic Body-Analysis Module (ST-GCN)

This module classifies aggressive behavior from the temporal dynamics of movement, rather than analyzing static frames.

4.3.1. Skeleton Normalization

33 keypoints per frame are extracted using MediaPipe Pose, each with (x, y, z) coordinates and a visibility channel. To make the classifier invariant to the subject’s position and distance relative to the camera, the skeleton is centered by subtracting the hip centroid (translation invariance) and normalized by dividing by the maximum spatial spread (scale invariance):

$$\tilde{\mathbf{p}}_i = \frac{\mathbf{p}_i - \mathbf{p}_{\text{pelvis}}}{d} \quad (4)$$

4.3.2. Classifier Architecture

The model processes an input tensor of dimensions (batch, 4 channels, 15 temporal frames, 33 nodes) through three stacked spatiotemporal blocks with residual connections: the first expands the 4 native channels to 64; the second widens the latent space to 128 channels and compresses the temporal dimension with a stride of 2; the third consolidates the representation into 256 channels. A global pooling layer averages the temporal and nodal dimensions, and a final 1×1 convolution projects the logits toward the binary classification (normal behavior versus threat).

Each spatial graph convolution is defined as $Z = A \times W$, where X is the input feature matrix, W the learnable weights, and A the adjacency matrix, symmetrically normalized via the diagonal degree matrix D to prevent gradient instability:

$$A = D^{-1/2} (A_{\text{raw}} + I) D^{-1/2} \tag{5}$$

This module’s execution is deliberately forced onto the server’s CPU: since the skeletal tensors are numerically sparse and lightweight (~7.9 KB per sequence), inference resolves in under 1 ms, freeing up the entire GPU video memory (VRAM) for the facial module’s dense convolutional networks.

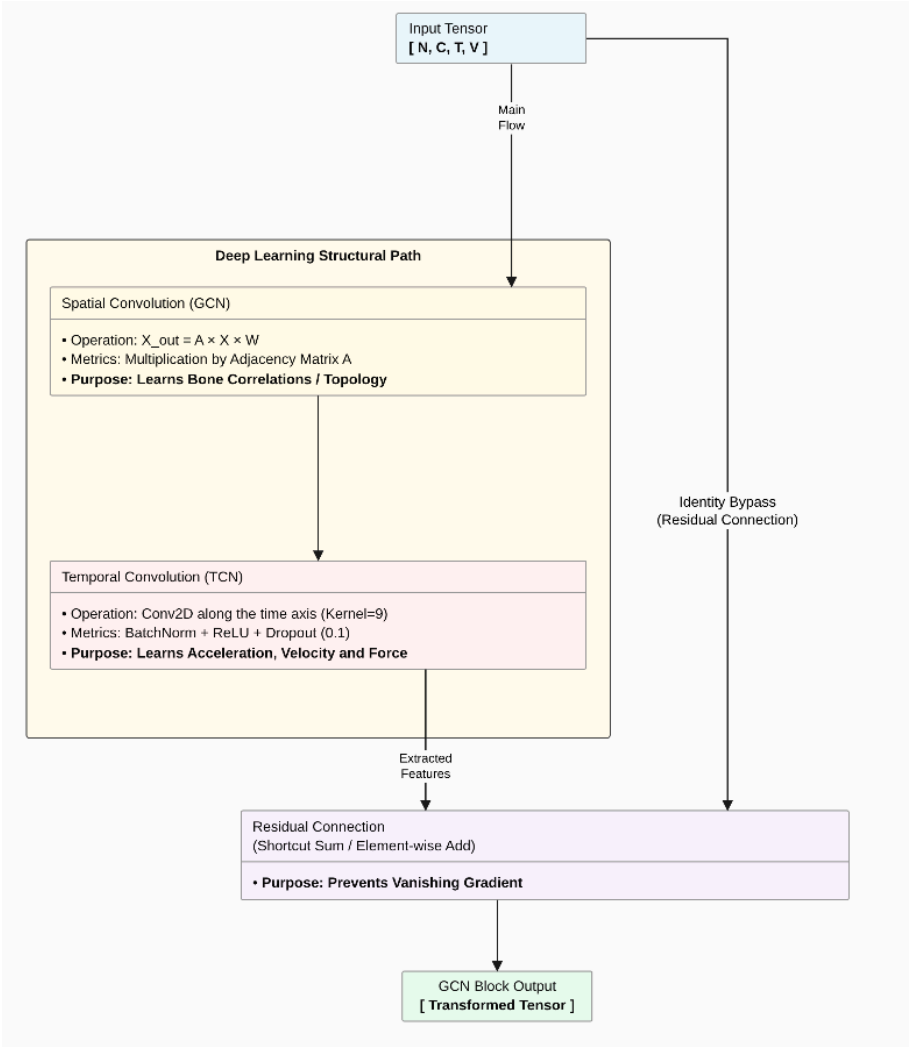


Figure 8. Internal architecture of the spatiotemporal convolution block.

4.4. Weapon Detection Module (YOLOv10 + ByteTrack)

4.4.1. Data and False-Positive Mitigation

The detector was trained on four public Roboflow Universe datasets [19], remapped to two classes: knife and pistol. To reduce false positives on elongated everyday objects (remote controls, phones, tools), two techniques were applied: injecting 15% negative images of empty residential backgrounds into each dataset split, and a second hard-negative-mining phase, in which the false positives and false negatives of the initial model were captured on an external validation set to build a refinement corpus.

The refinement combined the full set of mined hard cases with a 30% anchor sample from the original dataset (an experience-replay strategy, to prevent catastrophic forgetting), using the AdamW optimizer with a reduced learning rate and mixed precision.

4.4.2. Tracking and Temporal Confirmation

The bounding boxes emitted by YOLOv10 are fed into the ByteTrack algorithm, which associates detections over time via a track identifier. To filter out transient one- or two-frame false positives, each track is evaluated over a 1.5-second sliding window, requiring the simultaneous satisfaction of two conditions: a minimum number of confirmed detections ($N_{\text{hits}} = 5$) and a minimum average confidence (0.60) within that window:

$$\text{Confirmed threat} = (M \geq N_{\text{hits}}) \wedge \left(\frac{1}{M} \sum_{i=1}^M c_i \geq \bar{c}_{\min} \right) \quad (6)$$

4.5. Correlation Engine and Business Rules

The probabilistic outputs of the three modules converge in a central orchestrator that translates them into actionable security decisions via a small set of business rules. The most relevant are:

- Authorized-access threshold: an individual is considered an authorized resident if the cosine distance between their embedding and the stored reference satisfies $D_{\text{cos}} \leq 0.61$.
- Intruder detection (Gatekeeper): if no match reaches the above threshold, the identity is forcibly transmuted to “unknown person.”
- Sustained aggression: an aggression alert is triggered if the body classifier reports actions such as hitting, kicking, or struggling continuously throughout the evaluation time window.

The highest-priority rule is the threat-fusion rule, which combines the outputs of the facial module with those of the body or object module. The engine triggers a critical alert only when the detected identity is an unknown person and, simultaneously, the presence of a weapon or an aggressive action is confirmed within the same spatiotemporal window:

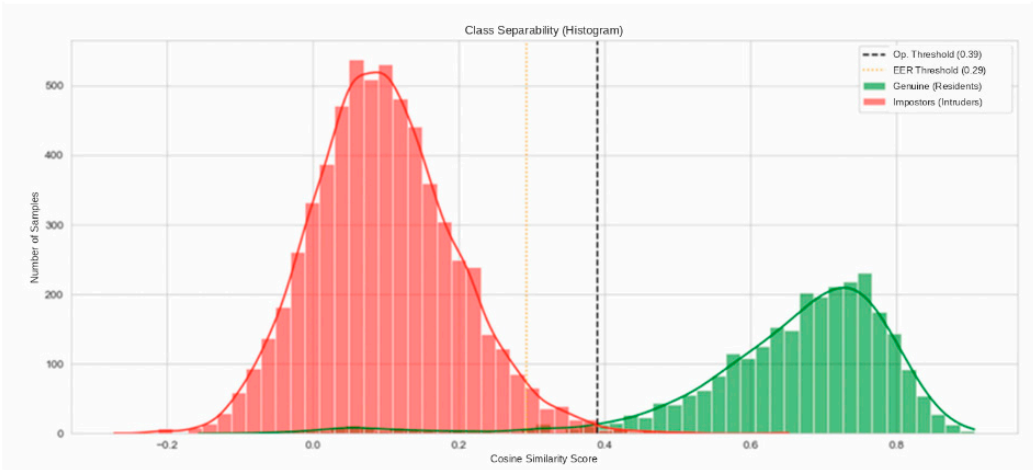
$$\text{Critical alert} = (\text{person} = \text{“Unknown”}) \text{ AND } (\text{action} = \text{“Aggression”}) \text{ OR } (\text{detection} = \text{“Weapon”}) \quad (7)$$

This composite logical condition is, in essence, the prototype’s differential contribution relative to the single-criterion systems described in Table 1: no module on its own—neither the facial, the body, nor the object module—generates a top-priority alert; only the coincidence of two independent signals in time and space enables it, which reduces the probability of critical false alarms and concentrates the operator’s attention on the events of greatest real risk.

5. Results

5.1. Facial Module and Biometric Calibration

The class-separability histogram obtained on LFW [18] shows two distributions—impostors and genuine users—intersecting at a cosine similarity of 0.29 (Equal Error Rate point). Maximizing the F1-Score shifted the operating cutoff point to a similarity of 0.39, i.e., $D_{\text{cos}} \leq 0.61$, which reduces the false-acceptance rate to a value close to zero. The nearest-neighbor query on pgvector’s HNSW index, with $\text{ef_search} = 32$, maintained a constant latency at or below 5 ms.



Threshold Selection

Figure 9. Biometric class distribution.

5.2. Kinematic Body Module

The ST-GCN classifier was evaluated on the skeletal modality of the NTU RGB+D dataset [17], using a weighted cross-entropy loss function (weight 3.0 for the physical-aggression class). Table 5 summarizes the metrics obtained.

Table 5. Performance metrics of the body module (ST-GCN).

Action class	Precision	Recall	F1-score
Everyday / noisy actions	0.93	0.91	0.92
Physical aggression / active threat	0.78	0.84	0.81

The loss-function weighting prioritized reducing false negatives—that is, missing a real fight—over a test set of 13,744 sequences, achieving an overall accuracy of 0.89 (TP = 3,248, TN = 8,960, FP = 898, FN = 638), with a precision of 0.78, a recall of 0.84, and an F1-score of 0.81 for the active-threat class. The model, compiled in ONNX, ran in under 1 ms per 15-frame sequence on CPU.

5.3. Object Detection Module

Refinement through hard-negative mining and background-image injection (13% explicit back-grounds in each dataset split) was aimed at reducing false positives on everyday objects. Table 6 summarizes the final metrics obtained on the test set.

Table 6. Final metrics of the weapon detector (YOLOv10) on the test set.

Class	AP@0.5	Recall	False positives on back-grounds
Knife	0.82	91%	58%
Pistol	0.97	86%	42%
Global	0.890	93%	F1 = 0.87 ($\tau \approx 0.453$)

5.4. Global System Telemetry

Table 7. Global performance and resource-consumption metrics.

Metric	Observed value	Component / rationale
Ingestion rate (sensor)	Sustained 10.0 FPS	Decoupled ingestion (SPMC pattern)
Effective end-to-end throughput	7.0–8.0 FPS	Active conflation (ZMQ_CONFLATE=1)
Total notification latency	Sub-second	Capture → ZMQ inference → orchestrator → WebSocket

5.5. Discussion

The results confirm the technical viability of multimodal edge analytics for residential complexes. Message conflation with ZeroMQ resolved the temporal lag characteristic of cumulative FIFO queues, ensuring that the operator always observes the most recent situation. Combining ArcFace with pgvector eliminated the double-write problem by maintaining PostgreSQL’s transactional consistency, while the rigid cosine-distance threshold shielded the system against identity spoofing. Delegating skeletal-graph inference to the CPU proved to be a sound architectural decision: by freeing up all VRAM for the facial module, both models coexist without degrading overall throughput. Finally, the weapon detector’s temporal-confirmation strategy significantly reduced the probability of an isolated false positive escalating into a critical alert, preventing the operator from being overwhelmed with false alarms.

Compared with the systems in Table 1, the prototype’s differential contribution does not lie in individually outperforming each computer-vision module against the published state of the art, but in the spatiotemporal correlation of their outputs: neither DRISHTI nor SmartFace triggers a critical alert exclusively when an unrecognized identity and a risk pattern (weapon or aggression) coincide, within the same time window and on the same camera. This composite rule reduces the number of events reaching the human operator, prioritizing those most likely to represent a genuine threat.

The prototype also presents limitations that should be read alongside its metrics. Biometric validation was performed on LFW, an academic reference dataset, rather than on a purpose-built dataset of residents under real conditions from a condominium’s cameras (variations in angle, backlighting, face masks), so field performance may differ from that reported in Section 5.1. Similarly, the body classifier was trained and evaluated on NTU RGB+D, whose recordings differ in framing and person density from a real hallway or vehicle-access point. These gaps between the evaluation environment and the deployment environment are consistent with the stated operational limitations and reinforce the need for field validation before any commercial deployment.

6. Conclusions

This work designed, implemented, and evaluated the prototype of a distributed smart security system for residential complexes, with the following main conclusions:

- It is feasible to bring video-surveillance analytics comparable to those of government command centers into the private residential domain, using open-source software and optimizing the hardware budget through edge decoupling.
- The event-driven inference mesh over ZeroMQ guarantees fault isolation: an exception in an AI worker neither halts video ingestion nor crashes the web API.
- Centralized graphical rendering guarantees absolute pixel parity between the live stream and the evidence stored in object storage, preserving the chain of custody.
- The prototype met its stated objectives: effective throughput of 7–8 FPS with sub-second latencies, an accuracy of 0.89 in classifying aggressive behavior (recall of 0.84 and F1 of 0.81 for the active-threat class), a biometric false-acceptance rate close to zero, and an operational interface for real-time incident management.

6.1. Future Work

- Replace the ST-GCN backbone with spatiotemporal Vision Transformers to capture complex, long-term interactions among multiple people.
- Export the models to engines such as TensorRT or CoreML with 8-bit quantization, to scale the system to dozens of cameras per node.
- Implement automatic face-blurring for visitors who do not trigger an incident, to comply with biometric data-protection regulations.
- Integrate vision-language models to automatically generate textual descriptions of incidents and enable semantic search over the historical log.

Data Availability Statement: The public datasets analyzed in this study are available from the sources cited in the References ([17–19]).

Acknowledgments: This work was developed as part of Trabajo Terminal 2026-A165 at the Escuela Superior de Cómputo (ESCOM), Instituto Politécnico Nacional, Mexico City, Mexico.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average precision
CCTV	Closed-circuit television
DVC	Data Version Control
ENSU	National Urban Public Safety Survey
FPN	Feature pyramid network
FPS	Frames per second
GCN	Graph convolutional network
HNSW	Hierarchical Navigable Small World
LFW	Labeled Faces in the Wild
ONNX	Open Neural Network Exchange
SPA	Single-page application
SPMC	Single-producer, multiple-consumer
ST-GCN	Spatiotemporal graph convolutional network
VRAM	Video random-access memory

References

1. Instituto Nacional de Estadística y Geografía (INEGI). Comunicación Social: Encuesta Nacional de Victimización y Percepción sobre Seguridad Pública (ENVIPE) 2024. Available online: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2024/ENVIPE/ENVIPE_24.pdf.
2. Zamora, Á. Videovigilancia e inteligencia artificial: entre la utopía y la distopía. **2021**, 4(14), 8–38. <https://doi.org/10.57042/rmcp.v4i14.432>.
3. C5 de la CDMX. El C5 de la Ciudad de México. Available online: <https://www.c5.cdmx.gob.mx/dependencia/acerca-de/el-c5-de-la-cdmx>.
4. Microsoft. Autenticación facial de Windows Hello. 16 October 2024. Available online: <https://learn.microsoft.com/es-es/windows-hardware/design/device-experiences/windows-hello-face-authentication> (accessed on 10 April 2025).
5. Deng, J.; Guo, J.; Yang, J.; Xue, N.; Kotsia, I.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
6. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.-E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 43, 172–186. <https://doi.org/10.1109/TPAMI.2019.2929257>.
7. Fang, H.; Xie, S.; Tai, Y.-W.; Lu, C. RMPE: Regional Multi-Person Pose Estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2334–2343. <https://doi.org/10.1109/ICCV.2017.252>.

8. Fanuel, M.; Muthusamy, K.; Das, S. A Survey on Skeleton-Based Activity Recognition Using Graph Convolutional Networks. National Science Foundation Research Report, 2021. Available online: <https://par.nsf.gov/servlets/purl/10356217>.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), 2016.
11. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.
12. Bhat, I. Drishti: An AI-Powered Surveillance System. GitHub, 2020. Available online: <https://github.com/ishwar-bhat/drishti>.
13. Ministry of Defence, Government of India. Defence Institute of Advanced Technology Wins First Prize in Smart India Hackathon-2020. Press Information Bureau, 6 August 2020. Available online: <https://www.pib.gov.in/Pressreleaseshare.aspx?PRID=1643742>.
14. Smart India Hackathon. Smart India Hackathon 2020 Themes. 2020. Available online: <https://www.sih.gov.in/sih2020>.
15. Abba, S.; Bizi, A.M.; Lee, J.-A.; Bakouri, S.; Crespo, M.L. Real-Time Object Detection, Tracking, and Monitoring Framework for Security Surveillance Systems. *Heliyon* **2024**, e34922. <https://doi.org/10.1016/j.heliyon.2024.e34922>.
16. Innovatrics. SmartFace: Real-Time Facial Recognition Platform. 2024. Available online: <https://www.innovatrics.com/smartface>.
17. Shahrudiy, A.; Liu, J.; Ng, T.-T.; Wang, G. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis; Nanyang Technological University: Singapore, 2016. Available online: <https://rose1.ntu.edu.sg/dataset/actionRecognition/>.
18. Li, J. LFW Dataset. Kaggle, 2018. Available online: <https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>.
19. Roboflow Universe. Gun and Knife MVZAR Dataset v3. 2024. Available online: <https://universe.roboflow.com/333-9gp58/gun-and-knife-mvzar/dataset/3>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.