technical transactions

No. 2026/007

information and communication technology

Paradigm

AIRIS: A Real-Time Face and Object Detection System for Threat Monitoring

Ilona Anna Urbaniak
ilona.urbaniak@pk.edu.pl | https://orcid.org/0000-0002-1948-6501
Wiktoria Maria Kosek
wiktoria.kosek@student.pk.edu.pl |
https://orcid.org/0009-0006-2941-281X
Alicja Maria Kowalska
alicja.kowalska@student.pk.edu.pl |
https://orcid.org/0009-0008-7381-479X
Cracow University of Technology

Scientific Editor: Radosław Kycia, Cracow University of Technology
Technical Editor: Dorota Sapek, Cracow University of Technology Press
Typesetting: Anna Pawlik, Cracow University of Technology Press

Received: January 22, 2026
Accepted: February 25, 2026

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Competing interests: The authors have declared that no competing interests exist.

Citation: Urbaniak, I.A., Kosek, W.M., Kowalska, A.M. (2026). AIRIS: A Real-Time Face and Object Detection System for Threat Monitoring. *Technical Transactions, e2026007*. https://doi.org/10.37705/TechTrans/e2026007

Abstract

This study presents AIRIS (Advanced Intelligent Recognition & Interception System), a real-time personal security monitoring platform integrating computer vision and artificial intelligence for mobile threat detection. The system is based on a three-layer architecture comprising adaptive face detection, temporal tracking, and hazardous object recognition using deep learning models. The main contribution lies in system-level integration and engineering validation under realistic deployment constraints. Individual identification combines embedding-based recognition with position-based tracking, while temporal persistence algorithms assess presence duration to identify potential risks. The implementation employs multithreaded processing and graceful degradation mechanisms to ensure reliable real-time operation in a wearable–mobile configuration. Experimental evaluation demonstrates 87% trial-level detection success for hazardous object presentation trials, 91% alert correctness, and processing throughput of 5–10 FPS with 120–180 ms latency.

Keywords: personal security, wearable vision, face recognition, temporal tracking, object detection, YOLO, dlib, OpenCV, edge AI, mobile inference

1. Introduction

According to the Global Organized Crime Index 2023 report, published by the Global Initiative Against Transnational Organized Crime, organized crime worldwide continues to rise, with 83% of the global population living under conditions of high crime levels (Global Initiative Against Transnational Organized Crime, 2024). This trend underscores the urgent necessity for developing innovative technological solutions to support citizen security. Concurrently, the rapid advancement of computer vision technologies and artificial intelligence (AI) have opened new opportunities for creating intelligent real-time monitoring systems.

Visual security monitoring evolved from early closed-circuit television (CCTV) systems toward increasingly automated solutions. An early example was proposed by Brown and Brown (1969), who introduced a television-based surveillance system for residential protection. These early systems were limited to fixed locations and manual observation, restricting their applicability for continuous personal security.

Subsequent developments primarily focused on stationary surveillance infrastructures, emphasizing centralized processing and static camera deployments (Masui et al., 2012). Although these systems improved situational awareness in public and industrial spaces, they remained unsuitable for individualized, mobile protection scenarios, particularly under constraints of limited computational resources and real-time responsiveness. This limitation highlighted the necessity for mobile security solutions that could accompany individuals throughout their daily activities (Masui et al., 2012).

Traditional security systems, predominantly based on stationary industrial cameras and centralized processing units, are characterized by significant limitations in individual applications. Early approaches to security surveillance relied heavily on conventional computer vision techniques and manual image analysis, which limited their effectiveness and flexibility.

Mahdi et al. (2017) presented a real-time surveillance system based on face recognition. The system utilized the Viola-Jones algorithm for face detection employing Haar-like features and Integral Image. The system operated correctly under standard lighting conditions (300–400 lx), however, its effectiveness significantly decreased under variable lighting conditions – the detection rate was only 30% or less. The Kanade-Lucas-Tomasi (KLT) tracker was employed for face tracking, which identified characteristic points, while Principal Component Analysis (PCA) was used for face recognition. These were among the most advanced techniques applied in visual systems at that time.

The effectiveness of face recognition in the system introduced by Mahdi et al. was dependent on several factors, including lighting conditions (achieving nearly 100% accuracy under constant illumination), face position (approximately 80% accuracy for positions present in the database), and object distance from the camera (approximately 62.5% accuracy with variable image depth). The system architecture comprised three separate modules: detection, tracking, and recognition, which required stable and controlled operating conditions. While the system employed approaches based on manually designed features and PCA analysis, newer systems utilize deep neural networks (Lu et al., 2021). In particular, Convolutional Neural Networks (CNN) with multi-layer architecture have significantly improved face recognition effectiveness.

An exemplary CNN structure consists of two convolutional layers (C1 and C2) and two pooling layers (S1 and S2), arranged alternately (C1–S1–C2––S2), as illustrated in Fig. 1. The network was designed to be robust to image

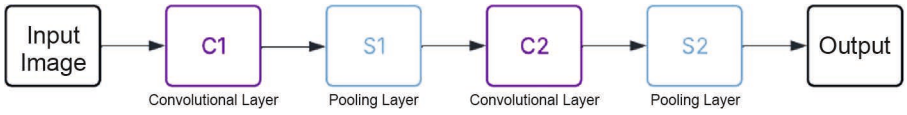

Fig. 1. CNN structure with alternating convolutional and pooling layers enabling robust image recognition under variable environmental conditions

https://doi.org/10.37705/TechTrans/e2026007

transformations such as translation, scaling, rotation, or horizontal flipping, enabling high recognition accuracy regardless of variable environmental conditions (LeCun et al., 1998).

With the advancement of AI algorithms, deep learning has become more accessible and easier to implement for individual developers, not exclusively for large corporations with substantial financial resources. Contemporary tools, libraries, datasets, and computational power are significantly more accessible, enabling AI to transition from a specialized technology into one accessible to a broad spectrum of users. For instance, open-source frameworks such as TensorFlow (Abadi et al., 2016), PyTorch (Paszke et al., 2019), and Keras (Chollet, 2015) have democratized deep learning development by providing high-level APIs and extensive documentation. Cloud-based platforms like Google Colaboratory (Google, 2020), Amazon SageMaker (Amazon Web Services, 2021), and Microsoft Azure Machine Learning (Microsoft, 2021) offer accessible computational resources for machine learning development. Furthermore, pre-trained models available through repositories such as Hugging Face Transformers (Wolf et al., 2020) and TensorFlow Hub enable developers to leverage state-of-the-art architectures without training from scratch. The availability of these advanced open-source frameworks, pre--trained models, and standardized APIs has revolutionized the landscape of AI application development, enabling the creation of sophisticated systems without the necessity of possessing extensive research infrastructure or considerable financial resources.

Despite these advances in AI technology accessibility, primary limitations of conventional systems include lack of mobility, prolonged response times, restricted accessibility, and absence of user personalization capabilities (Kyle, Lohn, 2023). In response to these challenges, portable monitoring systems are being developed that integrate advanced machine learning algorithms with wearable devices, offering more flexible, scalable, and individually tailored security solutions (Sabry et al., 2022).

Building upon this foundation, we have developed a personal security system called AIRIS (Advanced Intelligent Recognition & Interception System) based on real-time visual stream analysis. The presented implementation architecture encompasses two fundamental technological components that address the identified limitations of traditional approaches: face recognition and object detection, which operate in conjunction to enable monitoring of the user's environment.

Face recognition constitutes the first critical component of AIRIS, and it is specifically designed to recognize patterns of prolonged surveillance that may indicate a security threat. These algorithms utilize deep neural networks and biometric descriptors, facilitate not only the identification of individuals within the field of view but also enable temporal analysis of their presence.

Object detection represents the second core component, focusing on identifying potentially hazardous items within the user's vicinity. Contemporary deep learning architectures enable effective real-time object classification, which is crucial for ensuring immediate response to potential threats.

The integration of these two technologies within a single, mobile monitoring system represents an innovative approach to personal security challenges, combining advanced analytical capabilities of AI with the practical needs of contemporary society.

In this paper, we present AIRIS (Advanced Intelligent Recognition & Interception System), an integrated framework that combines adaptive face analysis, temporal persistence logic, and hazardous-object detection to support personal security monitoring in a wearable–mobile setup. The framework is validated through controlled experiments designed to reflect practical deployment constraints, including limited computational resources, variable illumination, and operation without external network connectivity.

Unlike standard pipelines that treat face recognition and YOLO as independent modules, AIRIS contributes a deployable, wearable–mobile architecture that couples identity persistence (embedding-based matching with centroid fallback), time-threshold risk logic, and controlled degradation under unavailable libraries or models. We validate this integration end-to--end in a real streaming setup (wearable camera → smartphone inference) under constrained conditions – including low light, operation without external network connectivity, and varying crowd density – reporting real-time latency and FPS together with alert correctness and detection accuracy.

This work therefore contributes an engineering-validated reference design for real-time, privacy-preserving personal monitoring, in which on-device constraints and failure modes (library availability, illumination variability, and loss of external connectivity) are treated as first-class design variables.

1.1. Contributions

This paper presents AIRIS (Advanced Intelligent Recognition & Interception System), an integrated framework for personal security monitoring in a wearable–mobile setting. The system combines adaptive face analysis, temporal persistence logic, and hazardous-object detection within a unified architecture.

The work emphasizes system-level design, integration, and performance evaluation of real-time AI components under practical deployment constraints, including limited computational resources and variable illumination. The framework is validated through controlled experimental evaluation reflecting these conditions.

1.2. Materials and Methods

This section presents the methodology and technical implementation details of the personal security monitoring solution. The system architecture and algorithmic approaches described herein were developed to address the critical need for real-time threat detection in personal safety applications, with particular emphasis on computational efficiency and deployment feasibility on resource-constrained edge devices in a wearable–mobile configuration.

The methodology encompasses several primary technical domains that collectively enable robust threat detection capabilities through an integrated approach. The foundational system architecture employs a three-layer design that separates data acquisition, processing, and presentation functionalities to ensure modularity and scalability across diverse deployment scenarios. A hybrid face detection approach implements adaptive algorithm selection between multiple computer vision libraries to maintain system reliability, while face identification methodology utilizes high-dimensional embedding vectors for robust facial recognition and tracking capabilities. The temporal persistence algorithm enables continuous identity tracking and threat assessment through robust monitoring of individual presence duration within the surveillance field, complemented by a dropout compensation mechanism that addresses transient detection failures through temporal logic approaches. Embedding stabilization under variable environmental conditions is achieved through moving average implementation, while hazardous object detection methodology utilizes advanced deep learning architectures for real-time threat identification. Comprehensive efficiency optimization techniques encompass multi-threaded processing architecture, adaptive control mechanisms, advanced image processing implementations, and computational complexity reduction strategies that collectively enable the system to achieve real-time threat detection performance while maintaining operational stability across diverse deployment environments.

https://doi.org/10.37705/TechTrans/e2026007

2. Three-layer system architecture

The implemented AIRIS system architecture was designed to ensure real-time processing capabilities while maintaining modularity and scalability. The system employs a three-layer processing architecture that enables efficient threat detection and monitoring.

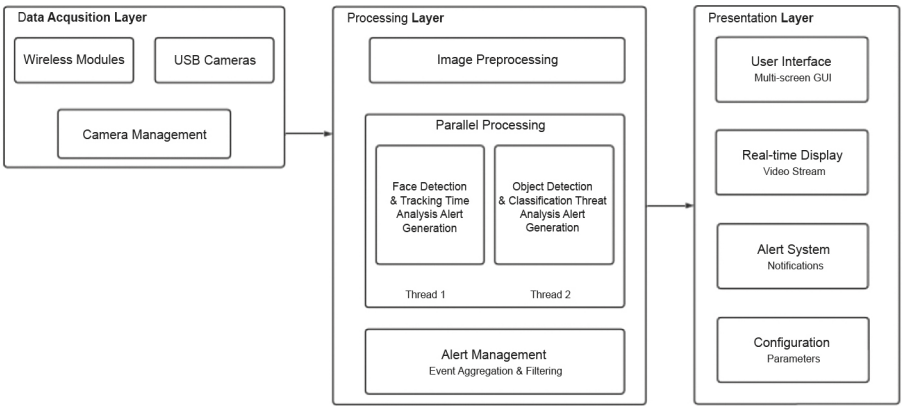

Fig. 2. AIRIS System Architecture

As illustrated in Fig. 2, the system comprises three distinct layers that provide functional separation between data acquisition, processing, and presentation. This architectural approach ensures optimal scalability and system maintainability while enabling independent operation of individual system components.

The Data Acquisition Layer implements visual stream acquisition functionality through a Camera Management module that handles both local USB cameras and wireless modules communicating via HTTP over a local network. In the evaluated wearable–mobile setup, the wireless module corresponds to a wearable camera unit streaming to the smartphone. This layer implements mechanisms for automatic detection of available recording devices, dynamic switching between image sources, and adaptive control of acquisition parameters such as resolution and FPS.

The Processing Layer is responsible for image analysis and threat detection. It implements visual preprocessing utilizing Contrast Limited Adaptive Histogram Equalization (CLAHE) techniques and filtering in the LAB color space. The primary components of this layer are the Face Detection and Object Detection modules, which execute detection algorithms in separate threads with frame buffering mechanisms and adaptive processing frequency control. The Face Detection module implements face detection and tracking algorithms utilizing dlib and OpenCV libraries, while the Object Detection module performs hazardous object detection based on a modified YOLO architecture.

The Presentation Layer was built using the Kivy framework and is responsible for implementing a cross-platform user interface. This layer provides functionalities for real-time visual stream display, a hierarchical alert system, and a configuration parameter management module.

2.1. Face detection: a hybrid approach with adaptive library selection

The implemented system employs a practical face detection framework that incorporates an adaptive library selection mechanism with automatic fallback capabilities based on computational resource availability and library accessibility. This hybrid approach ensures optimal performance across diverse deployment scenarios while maintaining system reliability through redundant detection methodologies.

Primary Detection Engine: Implementation

The primary detection engine leverages the dlib machine learning library for facial recognition and landmark detection (King, 2009). During system initialization, the framework attempts to load the dlib library along with its core components: the dlib.get_frontal_face_detector() and the optional shape_predictor_68_face_landmarks.dat model. Upon successful library initialization, the system activates a configuration flag that enables access to dlib's advanced facial analysis capabilities, including high-precision face detection and 68-point facial landmark extraction. Through empirical observation and preliminary testing, this configuration was selected as it consistently delivered the best performance results in our experimental setup. The dlib-based approach demonstrated reliable detection capabilities and accurate landmark positioning across various facial conditions, including scenarios involving facial pose variations, partial occlusions, and varying lighting conditions, making it the optimal choice for the primary detection engine implementation.

Fallback Detection Mechanism: Haar Cascade Classifiers

In circumstances where dlib library importation fails due to dependency issues, computational constraints, or deployment environment limitations, the system automatically transitions to an alternative detection strategy utilizing OpenCV's Haar cascade classifiers. This fallback mechanism ensures continuous system functionality while maintaining acceptable detection performance levels.

The Haar cascade classifier implementation is grounded in the pioneering work of Viola and Jones, who introduced the "Rapid Object Detection using a Boosted Cascade of Simple Features" methodology (Viola, Jones, 2001). This machine learning-based approach constructs a cascade function through extensive training on positive image samples (containing facial features) and negative image samples (lacking facial characteristics). The resulting classifier demonstrates high computational efficiency in object detection tasks across diverse image datasets (Viola, Jones, 2001).

Technical Implementation of Haar Feature Extraction

Feature extraction involves computational analysis of rectangular regions within the image, where each Haar feature generates a scalar value through pixel intensity summation differences between predefined white and black rectangular areas (Viola, Jones, 2001).

This computational approach mirrors convolutional kernel operations, where feature values are obtained by subtracting the cumulative pixel intensity of black rectangular regions from the cumulative intensity of corresponding white rectangular regions. The cascade structure enables rapid elimination of non-facial regions during early classification stages, significantly reducing computational overhead while maintaining detection accuracy (Viola, Jones, 2001).

Adaptive Strategy Rationale

Although the Viola-Jones algorithm demonstrates lower accuracy compared to contemporary methods utilizing deep convolutional neural networks, its computational efficiency and compact memory footprint ensure continued relevance in resource-constrained environments (Viola, Jones, 2001). The algorithm's lightweight characteristics make it particularly suitable for embedded systems, mobile applications, and real-time processing scenarios where computational resources are limited.

The implemented hybrid approach capitalizes on the complementary strengths of both detection methodologies: dlib's superior accuracy and

https://doi.org/10.37705/TechTrans/e2026007

advanced feature extraction capabilities serve as the preferred detection engine, while Haar cascades provide a reliable backup solution ensuring system resilience across diverse deployment conditions. This architectural decision reflects a pragmatic balance between detection performance optimization and system reliability requirements in production environments.

2.2. Face identification with 128-Dimensional embeddings

The face identification module employs a pre-trained deep learning model that generates 128-D descriptor vectors, providing a robust foundation for facial feature extraction and recognition tasks.

As shown in Figure 3, the dlib face recognition model architecture comprises a ResNet network with 29 convolutional layers, essentially a modified version of ResNet-34 with several layers removed and the number of filters reduced to improve computational efficiency (King, n.d.). This architectural design maintains computational efficiency while preserving the essential feature extraction capabilities required for high-accuracy face recognition.

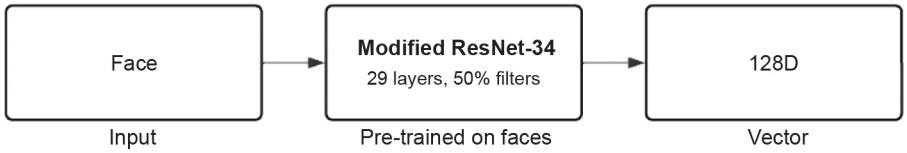

Fig. 3. Simplified ResNet architecture generating 128-Dimensional face embeddings

The system implements a robust methodology for creating aligned facial images through landmark-based geometric normalization, followed by the computation of a unique digital fingerprint for each detected face in the form of a high-dimensional numerical vector. The facial comparison process is realized through distance computation in multidimensional space, where the system measures the mathematical distance between two points representing faces in the 128-D embedding space. Each dimension in this space encodes specific anatomical facial characteristics, creating a detailed representation of facial geometry and texture patterns.

The underlying principle of this approach follows the embedding-based face recognition paradigm, which has become the standard methodology in modern face recognition systems since 2014. In this paradigm, smaller distances between points in the embedding space correspond to higher facial similarity (Schroff et al., 2015). These embeddings are stored in dedicated data structures for each actively tracked face, enabling rapid comparative analysis across subsequent video frames. When utilizing a distance threshold of 0.6, the dlib model achieves 99.38% accuracy on the standard LFW (Labeled Faces in the Wild) benchmark, demonstrating exceptional performance in identity recognition tasks under challenging real-world conditions.

2.3. Temporal persistence algorithm with identity tracking

The system implements face tracking through continuous monitoring of individual presence duration within the camera's field of view, utilizing dictionary-based data structures for efficient temporal information storage and retrieval. This approach enables tracking multiple faces throughout video sequences while maintaining identity associations.

Each detected face receives a unique identification code consisting of a sequential number and temporal detection timestamp, subsequently tracked through a parameter set describing its temporal presence characteristics. These parameters include: initial detection timestamp, last observation time, cumulative exposure duration, and alert status flag. The total presence time calculation employs a differential approach, computing the difference between

the current timestamp and initial detection time, while accounting for periods of temporary invisibility due to occlusions or head movements.

The system employs a dual-strategy approach for face matching between consecutive video frames: primary matching utilizes similarity comparison of digital fingerprints (128-D embeddings), while secondary matching employs geometric distance measurement between detected face region centroids. The geometric matching computes the Euclidean distance in horizontal and vertical coordinates, comparing this value against a threshold defined as half the larger dimension of the face bounding rectangle.

An automated threat detection mechanism generates alerts when the cumulative presence time of any individual exceeds a predefined temporal threshold, defaulting to ten seconds. This threshold-based approach enables the identification of potentially suspicious loitering behavior or unauthorized surveillance activities, providing an essential security monitoring capability.

2.4. Basic dropout compensation with temporal logic

The system implements a temporal dropout compensation mechanism for transient face detection failures without employing advanced predictive filtering techniques. This design choice prioritizes computational efficiency and implementation simplicity while maintaining effective tracking continuity under common real-world scenarios.

When a tracked face disappears from the camera's field of view, the system activates a specialized flag indicating a temporary disappearance state and records the precise timestamp of this event. Upon face reappearance, the system analyzes the duration of the detection gap through temporal logic evaluation. If the absence duration remains below three seconds – typically resulting from natural behaviors such as blinking, head rotation, or brief occlusions – the period is incorporated into the cumulative presence calculation, preserving temporal tracking continuity.

Individuals remaining invisible for periods exceeding ten seconds, indicating genuine departure from the monitoring area, are definitively removed from the tracking system to optimize memory resource utilization. While advanced tracking systems often employ mathematical motion prediction models or sophisticated Kalman filtering (Kalman, 1960) approaches for handling object occlusions, this implementation deliberately forgoes such complexity in favor of a simple yet effective temporal logic approach that accommodates natural human behaviors and typical visual system disturbances.

2.5. Exponential moving average for embedding stabilization

The system employs the mathematical technique of EMA for stabilizing digital face representations, effectively counteracting momentary fluctuations induced by environmental condition variations.

Exponential smoothing updates the current estimate using a weighted combination of the latest observation and the previous estimate, with exponentially decaying weights over time (Hyndman, Athanasopoulos, 2021). This characteristic makes EMA particularly valuable for applications requiring sensitivity to recent changes while maintaining historical context awareness.

The face representation update process operates through mathematical fusion of new measurements with previously averaged values, implementing a weighted combination where the system assigns 20% weight to new observations and 80% weight to the previous averaged representation. This weighting scheme creates a stable facial characteristic profile resistant to random variations while remaining adaptive to genuine appearance changes.

The underlying assumption of this technique is that more recent facial observations contribute more valuable information about an individual's true appearance compared to historical measurements, while simultaneously

https://doi.org/10.37705/TechTrans/e2026007

preserving accumulated knowledge about distinctive facial characteristics. The mechanism activates exclusively for faces already present in the system-newly detected individuals receive initial representations without modification, while subsequent observations of identical faces undergo the averaging process.

This approach results in progressive stabilization of each tracked person's digital fingerprint, enhancing recognition accuracy under variable illumination conditions, diverse viewing angles, and facial expression changes. The EMA--based stabilization effectively reduces noise in the embedding space while maintaining the discriminative power necessary for reliable face identification across challenging real-world scenarios.

2.6. Hazardous object detection

System Architecture

The foundation of the hazardous object detection system is a modified YOLOv4-tiny model (Bochkovskiy et al., 2020) employing a CSPDarknet-tiny convolutional neural network backbone, which represents a lightweight variant of the CSPDarknet architecture designed for resource-constrained real-time applications (Mahasin, Dewi, 2022). The neural network consists of twenty--nine convolutional layers containing 8.7 million parameters, which represents a practical balance between detection accuracy and computational efficiency for wearable devices requiring a balance between detection accuracy and computational efficiency.

The YOLOv4-tiny model was initialised from pre-trained weights available in the official Darknet repository (trained on the MS COCO dataset, 80 classes) and subsequently fine-tuned on a curated subset of publicly available images representing the 12 hazardous object categories targeted by the system (approximately 4,200 images across classes). A held-out validation split was used during fine-tuning to monitor convergence and calibrate operating thresholds. Class-specific confidence thresholds were determined through empirical tuning on the validation set: thresholds were set to minimise false negatives for high-risk items (e.g., knife, gun) while controlling false positives for contextually ambiguous dual-purpose objects (e.g., bottle, scissors), consistent with the risk-profile rationale described in Hurtado et al. (2021).

The hierarchical layer structure enables progressive acquisition of visual features at three levels of analysis. The initial layers detect basic visual descriptors such as object shapes, textures, and colors; the intermediate layers identify characteristic geometric patterns and object components; and the deep layers are responsible for object classification considering spatial context. With 8.7 million parameters, the model achieves high effectiveness in classifying twelve categories of hazardous objects while maintaining computational constraints for mobile systems.

The implementation of FPN (Lin et al., 2017) is responsible for detecting objects at different scales – from the smallest ones, such as a knife, to larger ones, such as a baseball bat. This is particularly important in the context of object detection in a dynamically changing camera field of view. The use of Leaky ReLU activation (Xu et al., 2015) throughout the entire network architecture contributes to improved gradients flowing through the network and enhancement of the learning process.

Tracking and Filtering

The AIRIS system implements spatial-based object tracking for temporal persistence between frames. The system uses simple spatial segmentation, creating tracking keys based on object position to associate detections across frames.

The implementation includes temporal result aggregation that reduces the number of false detections by analyzing detection stability in sequences of consecutive frames. A persistence counter tracks object stability, incrementing for consistent detections and decrementing when objects disappear. The persistent tracking mechanism confirms the actual presence of objects, requiring a minimum of two independent detections to eliminate false alarms. The system utilizes a queue storing the history of the last five detection results for each object, which enables assessment of detection consistency over short time intervals. Objects with persistence counters reaching zero are automatically removed to ensure efficient memory resource management.

2.7. Efficiency optimization

Multi-threaded Architecture

Parallel processing constitutes a key element of the AIRIS system's operational performance. The implementation of multithreaded architecture enables parallel execution of detection, analysis, and result presentation operations while ensuring consistent and secure access to shared system resources. The main thread is responsible for image acquisition and real-time result presentation to the user, while the background processing thread performs threat detection operations without blocking the visual interface.

The system achieves a user-interface update latency <25 ms, measured from the moment a detection result becomes available in the shared buffer to its on-screen presentation. Synchronization between threads is implemented using thread-locking mechanisms provided by the Python threading module.

The implementation of a frame buffering mechanism utilizing locks ensures secure access to shared data structures while maintaining high performance. The buffer management system maintains system operational stability.

Adaptive Control

The AIRIS system implements configurable processing frequency control through adjustable detection intervals. The system operates in two modes: standard mode with a detection interval of five frames and debugging mode with an interval of three frames. Mode selection is manually configured by the user through the debug toggle interface.

The reduced interval in debugging mode enables more frequent analysis for development and troubleshooting purposes, while the standard mode optimizes performance for regular operation. This simple yet effective approach balances detection responsiveness with computational efficiency based on user requirements.

Image Processing

Varied lighting conditions constitute a significant challenge for the effectiveness of real-time detection systems. In response to this issue, the system implements advanced image processing techniques based on the CLAHE algorithm (Zuiderveld, 1994) in the LAB color space, which significantly improves visibility under low-light conditions. A contrast limit (clipLimit) of 2.0 was applied, along with a tile grid size of 8 × 8 pixels (tileGridSize), ensuring adaptive local contrast enhancement without introducing artifacts in bright image areas. Processing in the LAB space enables independent optimization of the luminance component while preserving original chromatic information.

https://doi.org/10.37705/TechTrans/e2026007

Complexity Reduction

The system implements a series of computational complexity reduction techniques aimed at enabling efficient operation on resource-constrained devices. Implementation of exponential moving average with a smoothing factor $\alpha = 0.2$ enables efficient smoothing of detection results with minimal computational cost. The algorithm assigns greater weight to the most recent detection measurements while incorporating historical data to increase classification stability. The system automatically detects the availability of a CUDA-capable graphics processor and prefers GPU utilization for neural network processing, with automatic fallback to CPU processing when GPU availability is absent.

3. Experimental Validation: Methodology and Results

3.1. Methodology

To evaluate the performance of the AIRIS system, a testing scenario consisting of eight controlled tests was conducted, reflecting both typical operating conditions and boundary situations critical from the perspectives of reliability and safety. Each test focused on a different aspect of system functionality, including: lighting conditions, environmental dynamics, population density within the frame, no external network connectivity, effectiveness of hazardous object detection, and long-term stability. This arrangement enabled verification of both the system's fundamental functionality and its resilience to overload and environmental factors.

Test Environment and Equipment

- Hardware Configuration:
  The study employed an iPhone 15 smartphone with iOS 18.5, 128 GB from 2023, supplemented with a 10,000 mAh power bank for long-duration tests. Illuminance was measured using a digital lux meter (lx). In the tested deployment, video was acquired by a wearable camera module and streamed to the smartphone, which executed the detection pipeline and rendered alerts in the AIRIS interface. The wearable camera unit was an ESP32-WROVER-E CAM development board with an integrated camera, Wi-Fi, and Bluetooth Low Energy (BLE~4.2) connectivity (Fig. 4). For object detection tests, three elements representing typical categories of potentially hazardous objects were employed: a kitchen knife (blunted edge), scissors, and a plastic bottle.

Fig. 4. Wearable camera module used in the experimental evaluation: ESP32--WROVER-E CAM development board with integrated camera, Wi-Fi, and Bluetooth Low Energy (BLE~4.2) connectivity. The module served as an image acquisition and wireless streaming unit; all inference and decision logic were executed on the smartphone

- Software Configuration:
  AIRIS application version 1.3 was utilized, running in debug mode to log FPS, detection metrics, and diagnostic data. Logging was performed locally, with optional user-enabled cloud logging.

Test Scenarios

Eight experimental scenarios were prepared, organized into logical groups according to the phenomena under investigation:

- Illumination Variance Tests (T1–T3):
  - Controlled brightness environment (300–400 lx),
  - Low-light environment (<100 lx),
  - Outdoor environment under daylight conditions.

  Four repetitions were performed for each scenario.
- Population Density Tests (T4–T5):
  - Sparse environment with 1–3 individuals,
  - Crowded environment (≥ 4 individuals).

  The objective was to determine the impact of population count on FPS and detection quality.
- Network Independence Test (T6):
  The system was tested under complete network isolation (airplane mode). Each 15-minute trial was repeated three times.
- Object Detection Test (T7):
  Each object was presented to the system for 10 seconds in both static configuration and hand-held. Detection outcome (detected/not detected) and average classification confidence were assessed.
- Endurance Test (T8):
  System stability was monitored over 4 hours, recording parameters such as FPS, temperature, battery consumption, and memory usage at 30-minute intervals.

Evaluation Metrics

Primary evaluation indicators included:

- **FPS** – average number of frames processed per second;
- **Face detection accuracy** – percentage of correctly identified faces in the evaluated frames;
- **Trial-level detection success (%)** – percentage of 10-second object--presentation trials in which the correct object class was detected at least once at the specified confidence threshold (this metric does not correspond to mean Average Precision (mAP));
- **Alert correctness** – proportion of generated alerts that correspond to a confirmed threat event (either a face exposure exceeding the configurable time threshold or a hazardous object confirmed across the required number of consecutive frames) out of all alerts generated during a test scenario;
- **Detectability** – proportion of object-presentation trials in which the object was detected at least once;
- **Frame detection rate** – proportion of individual frames within a trial in which the object was correctly detected;
- **Temporal precision of alerts** – deviation from the 10-second threshold;
- **System stability** – number of crashes and rate of memory consumption growth. For network-dependent tests, transmission latency was also considered, while long-duration operation tests included battery discharge rate and device thermal behavior. Transmission latency over a standard WiFi 802.11n network was measured as 120–180 ms.

https://doi.org/10.37705/TechTrans/e2026007

All reported ± values correspond to standard deviation across repeated trials. Each experimental scenario was executed 3–5 times, and the reported statistics represent the variability observed across these repetitions.

Data Collection Procedure

Each test commenced with baseline measurements. During scenario execution, the system generated logs at regular 5-second intervals, recording both quantitative metrics and diagnostic messages. Following session completion, data underwent consistency verification and analysis. Each scenario was executed multiple times (3–5 trials, depending on scenario) to enhance result reliability. Reported ± values throughout the Results section correspond to standard deviation computed across repeated trials.

3.2. Validation results

Overall Performance

The AIRIS system successfully passed all eight tests (100% success rate). Table 1 presents the principal metrics, encompassing average frames per second, detection accuracy, and temporal alert concordance.

Table 1. Experimental test results of the AIRIS system

Test	Conditions	Mean FPS	Detection Metric [%]	Alert Correctness	False Positive Rate	Status
T1	Controlled illumination (300–400 lx)	8.2±0.4	94%	100%	0%	Active
T2	Low illumination (<100 lx)	6.1±0.8	87%	95%	5%	Active
T3	Daylight (outdoor)	7.5±0.6	91%	98%	2%	Active
T4	Sparse population (1–3 individuals)	8.8±0.3	96%	100%	0%	Active
T5	Dense population (≥ 4 individuals)	5.4±0.7	82%	89%	11%	Active
T6	No external network connectivity	8.1±0.5	93%	100%	0%	Active
T7	Hazardous object detection	7.8±0.4	87%	94%	6%	Active
T8	4-hour endurance	7.2 → 6.8	90%	97%	3%	Active

Note: For T1–T6 the detection metric corresponds to face detection accuracy; for T7 it corresponds to trial-level detection success

The false positive rate (FPR), computed as FPR = 1 − Alert Correctness, is also reported in Table 1 to provide a clearer assessment of system reliability. Across most scenarios the FPR remained low (0–6%), indicating that the alert generation mechanism effectively suppresses spurious detections. The highest FPR was observed in the dense population scenario (T5: 11%), which is consistent with the increased probability of transient detections in crowded environments. This behaviour is mitigated by the system's multi-frame persistence filtering mechanism, which requires detections to persist across consecutive frames before triggering an alert.

In illumination tests (T1–T3), the system maintained stable FPS and high detection efficacy, despite notable degradation under very low light conditions (25.6% FPS reduction). CLAHE implementation preserved functionality even below 100 lx. Population density scenarios demonstrated that AIRIS performs well with 1–3 individuals, whereas higher population counts reduce FPS to boundary values (5.4), though still compliant with requirements.

Hazardous Object Detection

Table 2. Hazardous object detection performance

Object	Confidence Threshold	Detectability	Mean Confidence	Frame Detection Rate
Knife	0.4	90%	0.72±0.11	90%
Scissors	0.3	75%	0.58±0.14	75%
Bottle	0.5	85%	0.81±0.09	85%
Mean	**variable**	**87%**	**0.70±0.13**	**83%**

Detectability indicates whether an object was detected at least once during a 10-second presentation trial, whereas frame detection rate reflects the proportion of individual frames within each trial in which the object was detected. Under the controlled test conditions used in this study, these values appear similar at the per-object level because successful trials typically produced detections across most frames. However, the reported mean values are computed across all trials and frames in the evaluation dataset; therefore, frame detection rate penalises partial-frame misses and may produce a lower aggregate mean than detectability.

Mean detectability reached 87%, with highest performance for knife detection (90%) and lowest for scissors (75%). These values indicate practical system utility while simultaneously identifying potential areas for further model enhancement.

Temporal Alert Correctness and Stability

Mean alert triggering time was 10.2±0.3 s (standard deviation across repeated trials; n = 3–5 per scenario). The mean delay is 0.2 s above the 10 s threshold (approximately 2%), indicating stable temporal presence analysis with low trial-to-trial variability. This result confirms the stability of the algorithm responsible for temporal presence analysis.

The 4-hour test revealed no crashes or operational interruptions. FPS gradually decreased from 8.1 to 6.8 (16% reduction), remaining within acceptable bounds. Memory utilization growth of 14% falls within the designed limit (<15%). Battery consumption totaled 45% (11.25%/hour).

4. Summary of Experimental Validation Results

Results confirm the practical utility of the AIRIS system under real-world deployment conditions. The 25.6% FPS decline in low illumination was anticipated; however, CLAHE implementation maintained functionality even below 100 lx, significantly expanding the operational range. Performance degradation in dense population scenarios (T5: 5.4 FPS) indicates the practical limit of current implementation on mid-range mobile hardware.

Trial-level detection success is reported as the percentage of 10-second object-presentation trials in which the correct object class was detected at least once at the specified confidence threshold. This metric reflects presentation--level detection reliability and should not be conflated with standard object detection metrics such as mean Average Precision (mAP).

The test T6 (No external network connectivity) confirms AIRIS suitability for privacy-sensitive environments.

Experimental validation confirms that the AIRIS system successfully achieves its design objectives:

https://doi.org/10.37705/TechTrans/e2026007

- 87% trial-level detection success (target: >85%)
- 5–10 FPS across varied conditions
- 91% temporal alert correctness
- 0 crashes in 4-hour test
- No external network connectivity functionality

The system demonstrates practical capability for real-world deployment in personal security applications on resource-constrained mobile devices.

4.1. System-level performance summary

System Performance Metrics

Detailed experimental protocols and per-scenario outcomes are reported in *Experimental Validation: Methodology and Results*. A concise summary of system performance is provided here to support interpretation.

As demonstrated by the experimental validation (*Experimental Validation: Methodology and Results*, Tables 1–2), the AIRIS system achieves reliable performance across key evaluation criteria. Hazardous object trial-level detection success reaches 87% under the class-specific confidence thresholds used in the evaluation (Table 2), indicating robust identification of potentially dangerous objects. Processing frequency is maintained within the range of 5–10 frames per second, enabling near real-time analysis suitable for practical security applications. Alert correctness reaches 91%, defined as the proportion of generated alerts corresponding to confirmed threat events (face exposure exceeding the time threshold or hazardous object persistence confirmation) when evaluated against predefined exposure-time thresholds.

Face Detection and Tracking Results

The implemented detection pipeline employs a two-stage approach combining Haar cascade classifiers and dlib deep learning models. This configuration enables the generation of 128-dimensional descriptor vectors for face identification, with automatic switching between OpenCV and dlib libraries when advanced recognition models are unavailable.

The temporal persistence algorithm effectively tracks individuals while compensating for brief detection interruptions of up to 3 seconds. The system monitors exposure duration and generates alerts upon exceeding a configurable time threshold (default: 10 seconds of continuous presence). False positive rates are reduced through spatial–temporal aggregation of detections, requiring multiple consecutive confirmations before classifying a presence as persistent. Dynamic management of tracked face objects supports efficient memory utilization and sustained computational performance.

Object Detection

The modified YOLO architecture classifies potentially hazardous objects using class-specific confidence thresholds derived from risk assessment profiles and contextual ambiguity (Hurtado et al., 2021). Detection thresholds are set to 0.3 for low-risk profile tools (e.g., scissors, fork), 0.4 for cutting tools and weapons (e.g., knife, gun, rifle, pistol, baseball bat, axe, hammer), and 0.5 for dual-purpose objects (e.g., bottle, wine glass, cell phone). Higher thresholds assigned to contextually ambiguous objects reduce false positives while preserving sensitivity for high-risk items.

Background processing enables continuous analysis without blocking the main application interface. A queue maintaining the most recent five detection results provides temporal smoothing and further reduces false alarms. Automatic removal of tracking elements after 10 seconds of absence from the field of view ensures efficient memory management.

System Architecture Validation

The three-layer system architecture operates in accordance with design specifications. Separation of data acquisition, processing, and presentation layers ensures modularity and independent operation of system components. The architecture supports integration with wireless camera modules for remote video streaming, while local USB cameras provide an alternative visual data source.

The multi-threaded implementation effectively separates acquisition device management, detection processes, and user interface functionality. In cases where YOLO models are unavailable, the system exhibits controlled functional degradation by switching to simulated detection mode while maintaining basic monitoring capabilities. Adaptive control of acquisition parameters, including resolution and frame sampling frequency, proves effective for performance optimization in mobile deployment scenarios.

Limitations, Ethics, and Privacy

The AIRIS system is designed with privacy preservation as a primary consideration. All processing is performed locally on the smartphone, with video transmitted only over a local wireless link (e.g., Wi-Fi between the camera module and the phone), without reliance on external networks or cloud services. Facial embeddings and detection results are stored only in volatile memory during active processing. Validation was conducted in controlled scenarios; real-world deployment requires informed user consent and raises ethical and privacy considerations to be addressed in future field trials.

4.2. Discussion

Comparative analysis with commercial solutions

To contextualize AIRIS relative to existing commercial wearable solutions, a qualitative comparison was conducted using publicly available product documentation for representative devices. The comparison focuses on system--level capabilities relevant to real-time personal threat monitoring, including on-device analytics, temporal reasoning, and architectural integration, rather than on general-purpose recording or augmented-reality functionalities. The results of this comparison are summarized in Table 3.

Table 3. Comparison of AIRIS with representative commercial wearable devices based on publicly available product documentation

Functionality	AIRIS System	Google Glass EE2 (Google, n.d.)	Axon Body-Worn (Axon, n.d.)
Primary application	Personal security monitoring	Enterprise / industrial AR	Law enforcement evidence capture
Face detection	On-device detection with identity persistence and exposure-time alerting	No built-in threat-monitoring face analytics described; requires third-party applications	Analytics depend on platform configuration; not described as on-device exposure-time alerting
Object detection	On-device hazardous-object detection with class-specific thresholds and temporal smoothing	No built-in hazardous-object detection described in official specifications	Product focus on capture and data management; real-time on-device hazardous-object alerts not described as core functionality
Image processing	CLAHE + LAB color space preprocessing	Standard device-level processing	Standard device-level processing
Architecture	Multi-threaded processing with frame buffering and graceful degradation	Wearable compute platform; application architecture is application-dependent	Capture-first architecture; analytics depend on external ecosystem components

As shown in Table 3, AIRIS differs from commercial wearable devices primarily in its system-level focus on real-time personal threat monitoring rather

https://doi.org/10.37705/TechTrans/e2026007

than general-purpose recording or augmented-reality applications. The AIRIS system integrates on-device face and object analysis with temporal reasoning mechanisms – such as exposure-time tracking and persistence filtering – that are not presented as built-in features in the examined commercial solutions.

While commercial devices offer robust hardware platforms and broad application ecosystems, the examined documentation emphasizes data capture and post-processing workflows or third-party application support. In contrast, AIRIS provides an integrated, privacy-preserving monitoring pipeline validated under wearable–mobile deployment constraints, supporting the development of a dedicated solution for personal security applications.

4.3. Future work

The future development of the AIRIS system will focus on several practical directions directly motivated by the experimental findings.

First, future work will explore behavioral pattern recognition as an extension of the current face and object detection framework. Building on the existing temporal tracking mechanisms, additional computer vision models could analyze movement patterns and spatial interactions to detect potentially threatening behaviours rather than relying solely on static object or identity recognition.

Second, the results of the dense population scenario (T5) indicate that crowded environments increase the probability of transient detections and false positives. To address this limitation, future development will investigate personalized exclusion algorithms that allow users to designate trusted individuals or familiar environments, enabling the system to suppress unnecessary alerts while maintaining vigilance for unknown threats.

Third, further work will focus on hardware miniaturization of the current ESP32-CAM prototype. Reducing the size and weight of the wearable module will improve usability and enable more practical integration into discreet wearable devices.

More advanced extensions, such as large-scale behavioral analytics or cloud-assisted services, remain potential long-term research directions but are outside the scope of the present prototype.

5. Conclusions

The AIRIS system demonstrates personal security monitoring capabilities in a wearable–mobile deployment, where a wearable camera streams video to a smartphone executing the detection pipeline. Performance evaluation indicates 87% trial-level detection success for hazardous object presentation trials and 91% alert correctness while maintaining processing rates of 5-10 FPS. The three-layer architecture integrates face detection, temporal tracking, and hazardous object detection through hybrid algorithms that balance computational efficiency with detection reliability. The temporal persistence algorithms enable monitoring of individual presence duration, while adaptive library selection mechanisms provide system resilience across deployment environments. Comparative analysis with commercial solutions indicates functional differences, with the AIRIS system providing specialized security monitoring capabilities not available in general-purpose wearable devices, enabled by its modular design and graceful degradation features.

The research contributes to understanding the deployment of computer vision algorithms on mobile platforms for personal security applications. The modular architecture and optimization techniques enable operation across varied hardware configurations, while the hybrid detection approach addresses limitations of individual algorithms. Areas identified for future development include behavioral pattern recognition, personalized exclusion algorithms,

and hardware miniaturization for practical deployment. Field validation across diverse populations remains necessary to establish system effectiveness in real-world conditions. Additionally, privacy and ethical considerations associated with continuous personal monitoring require further investigation. The results provide a foundation for future research in wearable security systems and demonstrate the potential for AI-enhanced personal protection applications in urban environments.

Featured Application and Practical Implications

The AIRIS system addresses critical gaps in personal security monitoring through real-time threat detection capabilities. The primary application focuses on identifying prolonged facial exposure (exceeding configurable time thresholds) and recognizing potentially dangerous objects in the user's environment, automatically generating security alerts when threats are detected. The system combines hybrid face detection with a modified object detection algorithm optimized for resource-constrained edge devices.

Beyond personal security applications, the modular architecture enables deployment across diverse domains including assistive technologies for individuals with disabilities, healthcare monitoring systems, and supervised care environments. The adaptive processing algorithms and performance degradation mechanisms support integration with existing surveillance infrastructure and mobile security platforms, while maintaining real-time operational requirements across varied hardware configurations.

Ethical and Privacy Considerations

The AIRIS system is designed with privacy preservation as a primary consideration. All visual data processing is performed locally on the mobile device, and the system does not rely on external network connectivity or cloud--based services for its core functionality. Video data are transmitted only over a local wireless link between the wearable camera module and the smartphone, without requiring internet access or remote servers, thereby limiting data exposure beyond the user's personal device ecosystem.

All face-related experimental testing was conducted exclusively with the voluntary and informed consent of the authors themselves; no third-party participants were involved in the validation study. Any future real-world deployment of the system in public or shared environments must be preceded by a formal privacy impact assessment and must comply with applicable legal and regulatory frameworks, including the General Data Protection Regulation (GDPR) in the European Union and equivalent national legislation.

Facial embeddings and detection results are maintained only in volatile memory for the duration of active processing and are not persistently stored or transmitted by default. The system is intended for personal security monitoring and requires informed user consent prior to deployment. Continuous visual monitoring inherently raises ethical considerations, particularly in shared or public environments; therefore, responsible and legally compliant deployment remains essential. Future work will include formal privacy impact assessments and field studies to further evaluate ethical implications under real-world deployment conditions.

Author Contributions

Ilona Anna Urbaniak, Wiktoria Maria Kosek, and Alicja Maria Kowalska contributed to the conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing (original draft preparation and review and editing), visualization, supervision, and project

https://doi.org/10.37705/TechTrans/e2026007

administration of this work. All authors have read and agreed to the published version of the manuscript.

Funding

This research was partially supported by the NAWA STARS EU BOOSTER project under the FERS.01.05-IP.08-0219/23 programme, pursuant to agreement no. BPI/WUE/2024/1/00029/U/00001.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability

Experimental data generated during system evaluation, including application logs and aggregated performance measurements, are available from the corresponding author upon reasonable request. Raw video streams and biometric data are not publicly shared due to privacy considerations.

Abbreviations

The following abbreviations are used in this manuscript:

AIRIS – Advanced Intelligent Recognition & Interception System
AI – Artificial Intelligence
YOLO – You Only Look Once
CNN – Convolutional Neural Network
FPS – Frames Per Second
CCTV – Closed-Circuit Television
CLAHE – Contrast Limited Adaptive Histogram Equalization
EMA – Exponential Moving Average
SMA – Simple Moving Average
FPN – Feature Pyramid Network
GPU – Graphics Processing Unit
CPU – Central Processing Unit
API – Application Programming Interface
LFW – Labeled Faces in the Wild

Terminology and System-Specific Definitions

- Face detection vs. identification:
  Face detection refers to the localization of facial regions within an image frame, while face identification assigns a persistent identity to detected faces using learned embedding representations.
- Embedding (128-D):
  A fixed-length numerical vector representing facial characteristics, used for similarity comparison and identity matching across frames.
- Digital fingerprint:
  A stabilized embedding representation associated with a tracked individual, updated over time to improve robustness under varying environmental and imaging conditions.
- Temporal persistence:
  A tracking mechanism that monitors the duration of an individual's presence within the camera field of view in order to identify prolonged exposure or loitering behavior.
- Exposure-time threshold:
  A predefined temporal limit (default: 10 seconds) beyond which a tracked individual is classified as potentially suspicious.

- Dropout compensation:
  A temporal logic mechanism that preserves identity continuity during short-term detection failures caused by occlusions, pose changes, or illumination variations.
- Persistence counter:
  A numerical indicator tracking detection stability across consecutive frames, used to confirm object or face presence and suppress transient false positives.
- Confidence threshold:
  A class-specific minimum detector score required to accept a face or object detection, calibrated to balance sensitivity and false-alarm rate.
- Hybrid face detection:
  An adaptive strategy that prioritizes deep-learning-based face detection and automatically falls back to lightweight feature-based methods under constrained computational conditions.
- Graceful degradation:
  A system-level design principle enabling controlled transitions between advanced and fallback algorithms to maintain functional operation under limited resources.
- Wearable–mobile deployment:
  An operational configuration in which a wearable camera streams visual data to a mobile device responsible for real-time processing and alert generation.
- Frame buffering:
  A synchronization mechanism that temporarily stores video frames to enable parallel acquisition, processing, and presentation in a multithreaded architecture.

References

Abadi, M., Barham, P., Chen, J., et al. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283.

Axon Enterprise, Inc. (n.d.). *Axon Body 4: Product overview / specifications*. Available online: https://www.axon.com/products/axon-body-4 (accessed on 14 January 2026).

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv*. https://arxiv.org/abs/2004.10934 (accessed on 14 January 2026).

Brown, M.V.B., Brown, A.L. (1969). *Home security system utilizing television surveillance* (U.S. Patent No. 3,482,037). Available online: https://patents.google.com/patent/US3482037

Chollet, F. (2015). *Keras: Deep learning library for Python* (Computer software). Available online: https://github.com/fchollet/keras (accessed on 14 January 2026).

Global Initiative Against Transnational Organized Crime. (2024). *The Global Organized Crime Index 2023*. GI-TOC. Available online: https://globalinitiative.net/analysis/ocindex-2023/ (accessed on 14 January 2026).

Google. (2020). *Google Colaboratory*. Available online: https://colab.research.google.com/ (accessed on 14 January 2026).

Google. (n.d.). *Glass Enterprise Edition 2: Tech specs / overview (Google Support)*. Available online: https://support.google.com/glass-enterprise/ (accessed on 14 January 2026).

Hurtado, J.V., Mohan, R., Burgard, W., Valada, A., Rauschenbach, T. (2021). Confidence score: The forgotten dimension of object detection performance evaluation. *Sensors*, 21(13), 4350. https://doi.org/10.3390/s21134350

Hyndman, R.J., Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts.
Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering 82(1)*, 35–45.
King, D.E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research 10*, 1755–1758.
King, D. (n.d.). *dlib-models*. GitHub repository. https://github.com/davisking/dlib-models (accessed on 13 March 2026).
Kyle, M., Lohn, A. (2023). Onboard AI: *Constraints and limitations*. Center for Security and Emerging Technology. https://doi.org/10.51593/2022CA008
LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278––2324.
Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125.
Lu, P., Song, B., Xu, L. (2021). Human face recognition based on convolutional neural network and augmented dataset. *Systems Science & Control Engineering 9*(sup2), 29–37.
Mahasin, M., Dewi, I. A. (2022). Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 backbones on YOLOv4 as object detector. *International Journal of Engineering, Science and Information Technology 2(3)*, 64–72.
Mahdi, F.P., Habib, M.M., Ahad, M.A.R., McKeever, S., Moslehuddin, A., Vasant, P. (2017). Face recognition-based real-time system for surveillance. *Intelligent Decision Technologies 11(1)*, 79–92.
Masui, K., Babaguchi, N., Dao, M., Mattivi, R., De Natale, F.G.B. (2012). A hybrid mobile-fixed surveillance system: A new solution for public security. Conference paper. https://doi.org/10.13140/RG.2.2.32158.02889
Microsoft. (2021). *Azure Machine Learning documentation*. Available online: https://docs.microsoft.com/en-us/azure/machine-learning/ (accessed on 14 January 2026).
Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems 32*.
Sabry, F., Eltaras, T., Labda, W., Alzoubi, K., Malluhi, Q. (2022). Machine learning for healthcare wearable devices: The big picture. *Journal of Healthcare Engineering*, Article 4653923. https://doi.org/10.1155/2022/4653923
Schroff, F., Kalenichenko, D., Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 815–823.
Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Vol. 1, I–I.
Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.
Xu, B., Wang, N., Chen, T., Li, M. (2015). Empirical evaluation of rectified activations in convolutional networks. *arXiv*. https://arxiv.org/abs/1505.00853
Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In *Graphics Gems IV*, 474–485.