Applied Computer Science 2026, vol. 22, no. 2, pp. 16–32
https://doi.org/10.35784/acs_8906

Submitted: 2025-12-22 | Revised: 2026-04-29 | Accepted: 2026-05-07

Keywords: object detection; face occlusion; YOLOv8; faster R-CNN; ATM security

Marco Manuel ARAGON PAUCAR [1], *Kelvin Yhonson FERNANDEZ ACERO* [1*], *Erasmo SULLA ESPINOZA* [1]

[1] Universidad Nacional de San Agustín de Arequipa, Perú, maragonp@unsa.edu.pe, kfernandezac@unsa.edu.pe, kfernandezac@unsa.edu.pe

[*] Corresponding author: kfernandezac@unsa.edu.pe

Detection of suspicious facial objects in neutral ATMs using deep learning architectures based on YOLOv8 and faster R-CNN

Abstract

This study presents an automated detection system for suspicious facial objects in neutral automated teller machines (ATMs) using two deep learning-based detection architectures: YOLOv8 and Faster R-CNN. The dataset was constructed from videos captured in ATM environments and complementary images associated with facial occlusion. From this material, frames corresponding to three classes (masks, hats, and sunglasses) were extracted, filtered, and annotated. Both models were trained using transfer learning and evaluated using standard object detection metrics, including precision, recall, F1-score, IoU, mAP@0.5, and mAP@0.5:0.95. The results show distinct behaviors between the two architectures. YOLOv8 achieved higher precision across the evaluated classes, reducing false positives and providing a more stable response to avoid unnecessary alerts. In contrast, Faster R-CNN achieved higher recall, showing greater sensitivity to partially visible facial objects, though with a higher tendency to generate false detections. Additionally, a deterministic rule was incorporated to classify each scene as NORMAL or SUSPECT, based on the number of valid suspicious classes detected and their temporal persistence in video sequences. The proposed system does not perform biometric identity recognition but instead focuses on identifying visual conditions associated with facial concealment. The findings suggest that deep learning-based object detection can support security in neutral ATMs through early visual alerts, although further validation in real operational environments is still required.

1. INTRODUCTION

Automating video surveillance systems with computer vision offers a support alternative for security in environments with limited supervision. Neutral automated teller machines (ATMs) operate autonomously and are typically located in public areas, which may expose them to situations involving coercion, in-person fraud, or facial concealment. In these scenarios, automatically detecting elements such as masks, hats, or sunglasses helps identify visual conditions that could hinder user identification during an operation.

Deep learning models have improved object detection in images and videos, enabling the localization of specific elements even under variations in lighting, pose, or partial occlusion. Single-stage architectures, such as YOLOv8, perform localization and classification within a single prediction flow, whereas two-stage models, such as Faster R-CNN, first generate candidate regions and then refine object classification and localization. This difference allows a comparison of both approaches in terms of precision, recall, spatial localization, and inference time.

This article presents a suspicious facial object detection system for neutral ATMs using YOLOv8 and Faster R-CNN. To this end, a dataset was constructed from videos captured in ATM environments and complementary images, considering three classes of interest: masks, hats, and sunglasses. Both models were trained and evaluated under a defined experimental configuration, and their results were compared using standard object detection metrics. In addition, a decision logic was incorporated to classify each scene as NORMAL or SUSPECT based on the combined presence of objects associated with facial concealment.

2. COMPUTER VISION

Computer vision encompasses techniques for processing images and videos to extract useful information about the elements present in a scene. In video surveillance systems, this area enables the detection of objects, faces, or accessories that may be associated with a risk condition. In neutral automated teller machines (ATMs), the application is relevant for identifying elements that partially cover the user's face, such as masks, hats, or sunglasses.

Deep learning-based object detection models are typically grouped into two categories. Single-stage detectors, such as YOLO and SSD, perform object localization and classification in a single process. Two-stage detectors, such as Faster R-CNN and Mask R-CNN, first generate candidate regions and then classify the detected objects. This distinction allows comparison of model behavior in terms of precision, sensitivity, and spatial localization. To evaluate their performance, metrics such as mean Average Precision (mAP) and Intersection over Union (IoU) are used, together with annotated datasets such as COCO and PASCAL VOC (Mittal, 2024).

Recent studies also highlight the usefulness of transfer learning and few-shot learning, especially when the available dataset is limited. Wu et al. (2025) propose combining global and local information to improve model generalization for new classes. Similarly, Zhang and Gu (2023) point out that reusing pretrained layers facilitates model adaptation to specific tasks without requiring large volumes of data.

The application of computer vision in real-world environments requires accounting for variations in lighting, scale, face position, and partial occlusion. Therefore, this study employs detection models oriented toward recognizing small or partially visible facial objects. The comparison between YOLOv8 and Faster R-CNN allows evaluation of their performance in detecting masks, hats, and sunglasses within images associated with neutral automated teller machines (Mittal, 2024).

3. DEEP LEARNING

Deep learning is a branch of machine learning that enables training models to recognize patterns directly from data. Unlike traditional methods, where features are usually manually defined, deep models progressively learn visual representations, facilitating tasks such as image classification, facial recognition, and object detection (Trigka & Dritsas, 2025).

In computer vision, convolutional neural networks enable the extraction of visual features at multiple levels, ranging from edges and textures to more complex shapes. This capability allows objects to be localized within images or videos, even under variations in lighting, scale, or position (Elrahman et al., 2025). Therefore, deep learning has been applied to surveillance systems, image analysis, and automatic detection of visual elements of interest (Makhlouf et al., 2024). Models that combine spatial and temporal information to analyze video sequences have also been developed.

3.1. Faster R-CNN

The Faster R-CNN (Regions with Convolutional Neural Networks) model is a widely used object detection architecture for its efficiency and effectiveness. It is a two-stage model. In the first stage, a Region Proposal Network (RPN) generates candidate regions of interest from the image; the second stage classifies these regions and refines the coordinates of the detected objects. This method enables precise localization and recognition of objects in images or videos while minimizing external computation and maximizing inference speed (Xiao et al., 2020).

Likewise, Kim et al. (2021) confirm that the Faster R-CNN architecture begins by extracting visual features from convolutional layers, which generate several feature maps that characterize specific image patterns. The feature maps are processed through the RPN module, which proposes candidate regions that may contain objects. The authors propose the RGDiNet model as an efficient Faster R-CNN variant for embedded systems. From this architecture, a real-time detection capability for low-power consumption datasets can be inferred.

With respect to security applications, Ingle et al. (2022) used Faster R-CNN to detect weapons in video surveillance environments, obtaining favorable results in identifying dangerous objects such as knives and firearms. Their findings indicate that the architecture remains accurate under different lighting conditions and in scenes containing multiple objects. In this regard, we implemented Faster R-CNN to improve weapon detection in closed-circuit television (CCTV) recordings. Similar to the previous case, they established a

surveillance image dataset containing weapons and demonstrated that the network enables efficient image segmentation and classification in urban scenes with high visual density.

Regarding industrial applications, Li et al. (2023) proposed a combination of Faster R-CNN and LSTM networks to detect unsafe behaviors at construction sites. In this approach, Faster R-CNN was responsible for spatial detection of workers and objects within the scene, while the LSTM network analyzed the temporal sequence of movements, achieving significant improvements in precision and a reduction in false positives.

On the other hand, Duong et al. (2023) carried out a review of deep learning-based anomaly detection methods, emphasizing the use of Faster R-CNN due to its good performance and its ability to maintain precision under complex conditions, such as variations in lighting interpreted as deviations from normal behavior or environments with many objects. The reviewed literature indicates that Faster R-CNN is a reliable and flexible architecture capable of accurately detecting behaviors and different types of objects within a video stream. Its architecture, as shown in Fig. 1, includes image input, feature extraction, region proposal generation, and final classification and localization.

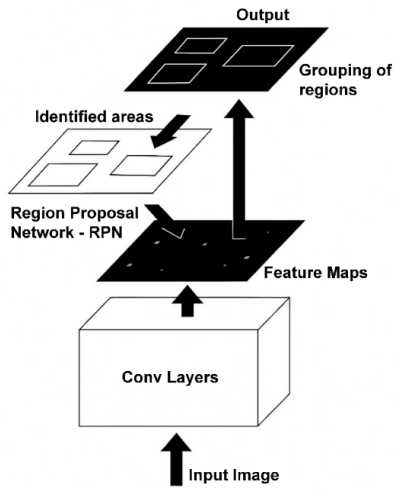

Fig. 1. Architecture of Faster R-CNN

3.2. YOLOv8

The YOLO (You Only Look Once) algorithm is a leading architecture for real-time object detection in computer vision. Introduced by Redmon et al. (2016), it unifies object localization and classification into a single stage, casting object detection as a direct regression problem mapping pixels to bounding box coordinates and their corresponding class probabilities (Hussain, 2023; Terven et al., 2023).

From its first version to YOLOv8, the architecture has evolved to improve precision, efficiency, and implementation. Key improvements include the use of C2f and SPPF modules, an anchor-free structure, and a decoupled detection head that optimizes coordinate and class prediction (Khalili & Smyth, 2024).

These advances enable YOLOv8 to achieve a strong balance between speed and accuracy, making it applicable in medicine, industry, and behavioral research. For example, Hermens (2024) reported that YOLOv8 maintains high performance even with small datasets, while Ju and Cai (2023) observed that this variant can also detect pediatric bone fractures with state-of-the-art results.

Therefore, YOLOv8 has become one of the most efficient object detection algorithms for real-time applications, offering good generalization alongside low computational cost. Developed by Ultralytics in 2023, YOLOv8 represents a significant advancement over previous versions by prioritizing real-time object detection with high precision and lower computational complexity (Yaseen, 2024; Khalili & Smyth, 2024). The architecture is organized into three main modules: Backbone, Neck, and Head, as shown in Fig. 2.

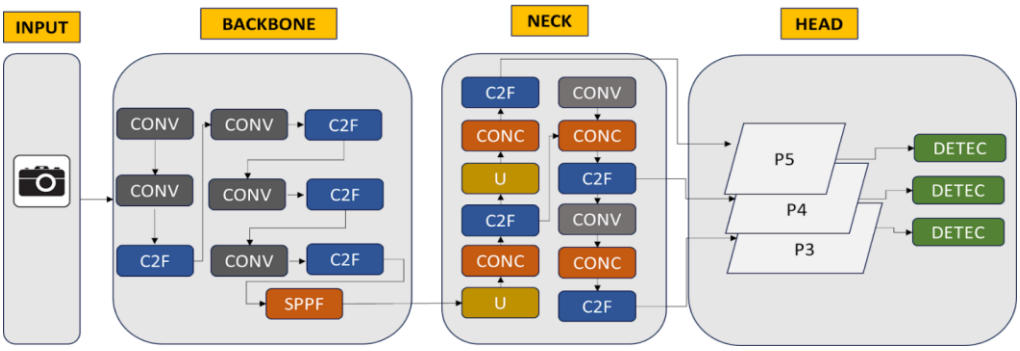

Fig. 2. Block diagram of the YOLOv8 architecture

4. SUSPICIOUS FACIAL OBJECT DETECTION

Face detection is an image-processing procedure that identifies and frames facial regions within an image. Feng et al. (2022) indicate that most contemporary systems are based on convolutional neural networks, which process images at multiple scales to extract facial features. These networks analyze facial shape through patterns of edges, textures, and contrast, enabling detection of faces despite variations in position and lighting. The same authors also argue that current models outperform earlier algorithms based on handcrafted classifiers because they enable large-scale learning.

Zhang et al. (2017) note that face detection is affected by partial facial occlusion, facial expressions, and the use of hair, masks, or glasses. Their study indicates that these conditions rarely leave a face fully visible and make it difficult to localize facial regions. The authors describe how deep convolutional networks generate feature maps at multiple scales to compensate for information loss and maintain detection performance in surveillance and video surveillance environments.

According to Thaer et al. (2025), in the context of automated security, facial detection systems rely on deep learning models to discriminate between visible faces and those affected by occluding elements. Their review highlights that subject detection rates depend not only on dataset quality but also on the variability of training scenarios.

Similarly, Rahim et al. (2023) state that combining convolutional neural networks (CNNs) with statistical models, such as logistic regression and gradient-based classifiers, increases the robustness of facial detection systems. These hybrid models improve discrimination between visible and partially covered faces, enhancing performance metrics such as precision, sensitivity, and F1-score.

Likewise, Selvi et al. (2022) propose the use of an enhanced convolutional neural network (ECNN) for the identification of suspicious actions in video surveillance environments. Their proposal, which incorporates the LeakyReLU activation function, improves face and gesture detection in complex situations involving both high crowd density and challenging lighting conditions.

On the other hand, Ihsan et al. (2025) designed an architecture for a real-time intelligent surveillance system based on the combination of YOLOv8 and DeepFace models, aimed at automatically identifying anomalous behaviors and facial emotions. Their research demonstrates that the detection of expressions such as fear, anger, or anxiety can support decision-making in security systems by combining facial recognition with motion pattern analysis.

Finally, Amirgaliyev et al. (2025) present a general review of machine learning and deep learning methods applied to facial recognition, highlighting the transition from classical approaches toward CNN-based and deep learning models.

5. NEURAL ATMS

Neutral automated teller machines (ATMs) are self-service financial terminals managed by banking institutions or municipal savings banks that allow users to perform transactions without in-person assistance. Unlike ATMs installed inside branches or supervised facilities, neutral ATMs are located in strategic locations such as shopping centers, gas stations, markets, or public spaces, where there is no direct personnel supervision. Their function is to expand the coverage of the financial system, ensuring that users can carry out transactions

such as withdrawals, deposits, or balance inquiries at any time through an interconnected network among different banking institutions.

Automated teller machines constitute an essential extension of modern financial services by enabling operations without human personnel. In the case of neutral ATMs, this principle is adapted to a decentralized environment in which the devices operate without physical assistance, but with remote monitoring through electronic security systems and discreet cameras, such as pinhole-type cameras employed in this research.

5.1. Vulnerabilities and Risks in ATM Environments

In addition to technological attacks such as skimming or PIN code inference through visual observation, neutral ATMs pose operational risks from coercion and direct crime. In documented incidents and police reports, offenders force users to withdraw money from nearby ATMs through threats or brief kidnappings; perpetrators may then move to other ATMs to carry out withdrawals, often concealing their faces with hats, masks, or other garments to evade video surveillance systems.

These scenarios combine physical and technical threats: facial concealment makes biometric identification more difficult and reduces the effectiveness of systems based solely on facial recognition. Therefore, the security of neutral ATMs requires multidimensional strategies that include, in addition to physical and software reinforcements, anomalous behavior detection, facial occlusion analysis, and temporal event correlation (for example, transaction sequences and movements between ATMs). Recent studies have shown that deep learning-based models can identify suspicious patterns in ATM usage before economic losses occur, and are especially useful for detecting atypical behaviors rather than identifying individuals in occlusion scenarios.

When addressing these risks, it is important to incorporate ethical safeguards: detection systems should anonymize images, restrict access to recordings, and alert authorized personnel while avoiding the disclosure of personal data, ensuring that improvements in security do not compromise user rights.

5.2. Integration of computer vision in neutral ATMs

The implementation of computer vision and deep learning models in ATMs has emerged as an effective technique for improving security. Recent studies have deployed architectures such as YOLOv8 and Faster R-CNN to detect individuals and unusual behaviors in real time within a framework that is both accurate and fast (Wu et al., 2019).

In this context, neutral ATMs become environments in which these techniques can be naturally applied, since their interoperable nature allows the integration of discreet pinhole-type cameras and autonomous monitoring systems.

The model presented in this research uses these cameras to capture users' facial characteristics during their interaction with the ATM and to detect suspicious conditions, such as the use of masks, hats, or sunglasses, which may inhibit biometric identification or suggest risk-related behaviors.

6. PROPOSED SYSTEM ARCHITECTURE

The proposed system focuses on detecting suspicious facial objects without performing biometric identity recognition, prioritizing user privacy. The acquisition and use of visual data are limited to research and security purposes at neutral ATMs, under data protection and restricted access criteria. Likewise, compliance with Peru's Personal Data Protection Law No. 29733 and the international privacy principles established in the European Union General Data Protection Regulation (GDPR) is considered.

The proposed system is structured into four main components:

1. image acquisition via a pinhole camera integrated into the neutral ATM;
2. preprocessing and normalization of the visual stream;
3. detection and classification of suspicious facial objects using YOLOv8 and Faster R-CNN;
4. automatic generation of an analysis state ("NORMAL" or "SUSPECT") to alert risk conditions.

Figure 3 shows the structural layout of a neutral ATM and the strategic placement of a pinhole camera integrated into the upper frontal section of the ATM within the ATM administrative area (ADM ATM). This type of concealed camera, used in banking video surveillance systems such as the XNV-6001 by Wisenet or the DS-2CD6425G1 by Hikvision, incorporates fixed lenses from 2.8 mm to 4 mm, with horizontal fields of view of approximately 75° to 108°, enabling frontal capture of the user's face at interaction distances of 1.5 to

3 meters. In addition, these cameras include high-sensitivity sensors for low-light conditions, facilitating the detection of suspicious facial objects without interfering with the normal operation of the ATM. The scheme also identifies the administrative data area (ADM DATA ATM), intended for equipment such as NVR, switch, router, UPS, and alarm panel systems, as well as the ATM vault where cash is stored.

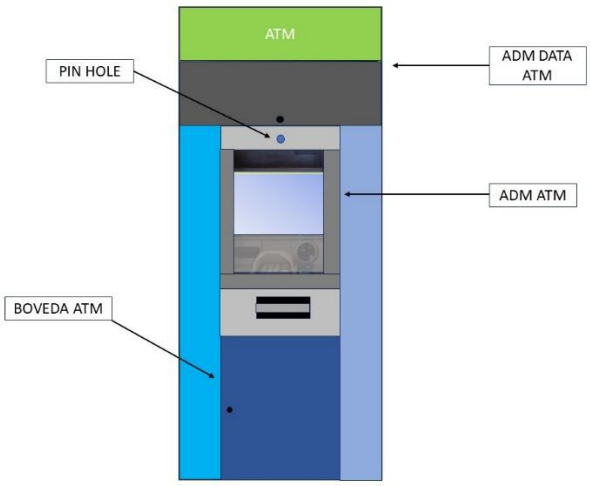

Fig. 3. Schematic view of the neutral ATM with a pinhole camera

Figure 4 illustrates the sequence of steps used during the training process. Frames are extracted from the input video and then preprocessed and filtered before being annotated based on the user's interaction with the ATM. This dataset is used to train the YOLOv8 and Faster R-CNN models, and the resulting architectures are evaluated using standard object detection metrics.

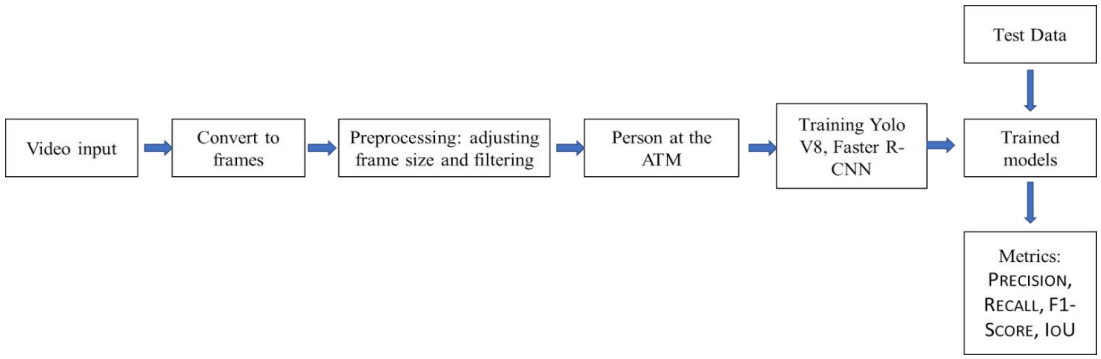

Fig. 4. Training architecture for detecting suspicious objects

Figure 5 shows the operational flow used during the system testing phase. Frames are extracted from the input video, preprocessed, and then analyzed by the YOLOv8 and Faster R-CNN models to detect facial objects such as masks, hats, or sunglasses. After extracting relevant features, the system classifies the scene into two possible states: NORMAL or SUSPECT.

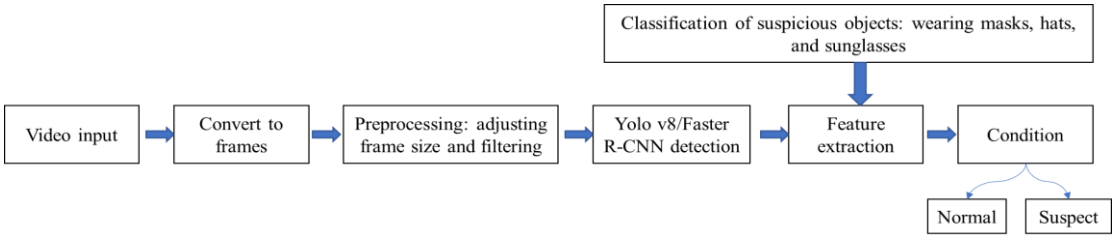

Fig. 5. Test architecture for detecting suspicious objects

7. EXPERIMENTAL PROCEDURE

The experimental procedure was developed in three stages: data preparation, model training, and system evaluation. Videos captured at neutral ATMs were converted into frames, filtered, and annotated for the classes masks, hats, and sunglasses. Subsequently, the YOLOv8n and Faster R-CNN ResNet50-FPN models were trained using transfer learning. Finally, their performance was evaluated using detection metrics and the final NORMAL/SUSPECT classification. The general configuration of each model is presented in Table 1.

Tab. 1. General experimental configuration of the evaluated models

Parameter	YOLOv8	Faster R-CNN
Architecture	YOLOv8n	Faster R-CNN ResNet50-FPN
Detector type	Single-stage	Two-stage
Framework	Ultralytics 8.3.228 + PyTorch 2.8.0	PyTorch + Torchvision
Programming language	Python	Python
Training type	Transfer learning	Transfer learning
Initial weights	yolov8n.pt	Faster R-CNN ResNet50-FPN pretrained on COCO
Main data source	Videos captured in neutral ATMs	Videos captured in neutral ATMs
Number of videos used	11 videos	11 videos
Video duration	Between 4:00 min and 5:30 min	Between 4:00 min and 5:30 min
Data conversion	Frame extraction from videos	Frame extraction from videos
Initial collected dataset	1400 images/frames	1400 images/frames
Preprocessing	Frame resizing, filtering, and cleaning	Frame resizing, filtering, and cleaning
Final training images	599	642
Test data	Validation frames and images; videos for functional testing	Validation frames and images; videos for functional testing
Validation images	57	57
Validation instances	143	141
Evaluated classes	Masks, hats, and sunglasses	Masks, hats, and sunglasses
Number of classes	3	3
Annotation format	YOLO TXT	PASCAL VOC XML
Annotation type	Bounding boxes	Bounding boxes
Epochs	30	30
Batch size	2	2
Input resolution	640 × 640 px	640 × 640 px
Device	CPU	GPU
Workers	1	1
Optimizer	Ultralytics automatic optimizer	SGD
Initial learning rate	0.01	0.005
Momentum	0.937	0.9
Weight decay	0.0005	0.0005
Patience / early stopping	10 epochs	10 epochs
Confidence threshold	0.25	0.50
IoU evaluation threshold	0.50	0.50
NMS / IoU	0.70	0.50
Evaluation metrics	Precision, recall, F1-score, IoU, mAP@0.5, and mAP@0.5:0.95	Precision, recall, F1-score, IoU, mAP@0.5, and mAP@0.5:0.95
System output	NORMAL/SUSPECT classification	NORMAL/SUSPECT classification

Initially, 1400 images and frames from videos captured in ATM environments, along with complementary images associated with facial occlusion, were collected. However, not all records were included in training, as a prior filtering stage removed blurred images, duplicate samples, images lacking a clear representation of the evaluated classes, and inconsistent annotations. As a result, the final dataset consisted of 599 images for YOLOv8 and 642 images for Faster R-CNN.

The difference in the final number of images for each model arose from the annotation conversion and validation process. YOLOv8 used the YOLO TXT format, whereas Faster R-CNN employed annotations in

PASCAL VOC XML format. During this stage, some images were discarded because they contained incomplete bounding boxes, incompatible labels, or no valid objects for training. Therefore, although both models originated from the same initial dataset, the final number of images was not exactly the same.

Training was performed using transfer learning with pretrained weights to reduce dependence on large volumes of data and facilitate adaptation to the study classes: masks, hats, and sunglasses. For YOLOv8, the lightweight YOLOv8n variant with yolov8n. The initial weights were used, whereas Faster R-CNN employed the COCO-pretrained ResNet50-FPN architecture. This approach reused visual features learned from large-scale datasets and adapted the final layers to the evaluated facial objects.

Both models were trained for 30 epochs, with a batch size of 2 and an input resolution of 640 × 640 pixels. The number of epochs was considered appropriate given the moderate size of the training dataset and the use of transfer learning, avoiding excessive training that could lead to overfitting. The reduced batch size addressed memory limitations of the execution environment and allowed the same base configuration to be maintained for both models. The 640 × 640 pixel resolution was selected because it better preserves details of small or partially visible objects, such as sunglasses and masks, without altering the standard training workflow of detection models.

The difference in the training device was due to computational availability and the processing cost of each architecture. YOLOv8 was executed on CPU, whereas Faster R-CNN was trained on GPU because its two-stage architecture requires greater computational capacity for candidate region generation and subsequent classification. This difference should be considered when interpreting processing times, since the device directly influences training and inference speed.

For YOLOv8, the Ultralytics automatic optimizer was used with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. Faster R-CNN employed SGD with an initial learning rate of 0.005, momentum of 0.9, and weight decay of 0.0005, a configuration commonly used in models based on PyTorch and Torchvision. In both cases, a patience value of 10 epochs was used to control overfitting and stop training if no improvement in performance was observed.

During inference, different confidence thresholds were used. YOLOv8 employed a threshold of 0.25, a common value in YOLO-based models to avoid discarding early detections, especially for small or partially occluded objects. Faster R-CNN used a threshold of 0.50 to filter low-confidence detections and reduce false positives. For evaluation, an IoU threshold of 0.50 was applied as the criterion for considering a detection correct when sufficient overlap existed between the predicted bounding box and the ground-truth box. Likewise, non-maximum suppression was applied to eliminate duplicate boxes for the same object.

Figure 6 formalizes the statistical and operational rules used to convert YOLOv8 and Faster R-CNN detections into a final NORMAL/SUSPECT output. The process begins with inference on an image or video frame, where each model generates bounding boxes, detected classes, and confidence levels for three classes: masks, hats, and sunglasses.

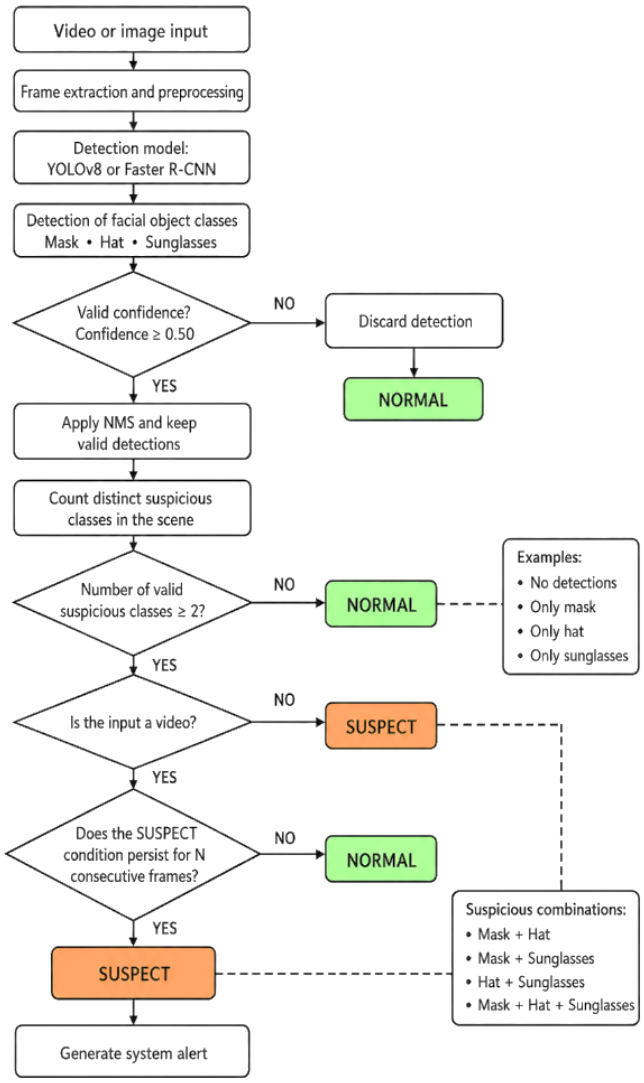

Fig. 6. Decision logic for NORMAL/SUSPECT classification

To reduce weak detections, only predictions exceeding the defined confidence threshold are considered valid. In the flowchart, a minimum confidence threshold of greater than or equal to 0.50 is used as the detection acceptance criterion. Subsequently, NMS is applied to eliminate duplicate bounding boxes for the same object, keeping only the detection with the highest confidence. This prevents the system from counting the same class multiple times.

The decision is based on a deterministic class-counting rule. If the number of valid suspicious classes is 0 or 1, the scene is classified as NORMAL. If the number of valid suspicious classes is greater than or equal to 2, the scene is classified as SUSPECT. Therefore, a single detection, such as only a mask, only a hat, or only sunglasses, does not trigger the alert, since it may correspond to a normal condition or a false detection.

Statistically, the rule can be expressed as follows:

$$S = B + G + L \tag{1}$$

Where:

$$B = \begin{cases} 1, & \text{if a mask is detected with confidence } \geq 0.50 \\ 0, & \text{if a mask is detected or the confidence is } < 0.50 \end{cases}$$

$$G = \begin{cases} 1, & \text{if a hat is detected with confidence } \geq 0.50 \\ 0, & \text{if a hat es detected or the confidence is } < 0.50 \end{cases}$$

$$L = \begin{cases} 1, & \text{if sunglasses are detected with confidence } \geq 0.50 \\ 0, & \text{if no sunglasses is detected or the confidence is } < 0.50 \end{cases}$$

The final classification of an image is defined as:

$$D = \begin{cases} NORMAL, & \text{is } S < 2 \\ SUSPECT, & \text{is } S \geq 2 \end{cases} \quad (2)$$

For video evaluation, a temporal condition is included. The scene is classified as SUSPECT only if the condition $S \geq 2$for N consecutive frames. This limitation helps reduce false alarms caused by isolated detections, user movement, lighting changes, or momentary model errors. Operationally:

$$D_{video} = \begin{cases} NORMAL, & \text{si } S_t < 2 \text{ does not persisnt for N frames} \\ SUSPECT, & \text{si } S_t \geq 2 \text{ during N consecutive frames} \end{cases} \quad (3)$$

In this way, the decision logic does not depend on subjective interpretation but on three measurable criteria: minimum detection confidence, the number of valid suspicious classes, and temporal persistence in video sequences. This rule remains consistent with the experimental procedure because it uses the trained classes, inference thresholds, and the evaluation process for images and videos defined in the study configuration.

8. YOLOV8 TRAINING AND RESULTS

This section presents the YOLOv8 training process and the resulting evaluation metrics. The model was trained on a dataset compiled from images captured at neutral ATMs and supplemented with data from public datasets. All images were annotated in Roboflow to identify the facial objects of interest. Finally, model performance was evaluated using standard metrics such as precision, recall, and mAP, enabling analysis of its detection capability under varying lighting and facial occlusion conditions.

Figure 7 shows the distribution of the dataset used to train the YOLOv8 model. The upper-left section presents the number of instances per class, with masks showing the highest frequency, followed by hats and sunglasses. The upper-right section displays the overlap of annotated bounding boxes, illustrating variability in the size and position of the labeled objects. The lower graphs represent the spatial distribution of annotation coordinates and dimensions, providing an overview of the range of positions, heights, and widths of the objects within the dataset.

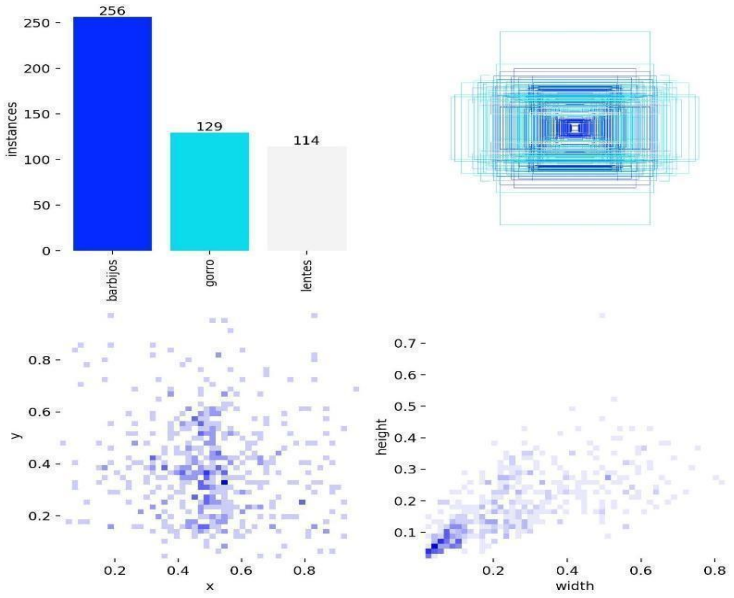

Fig. 7. Distribution of the dataset used for YOLOv8 training

Figure 8 presents visual examples of detections produced by the YOLOv8 model on various images from the test dataset. The system correctly identifies suspicious facial objects, such as masks, hats, and sunglasses, using bounding boxes and corresponding confidence scores. The model maintains consistent performance across variations in pose, lighting, and accessory type.

Fig. 8. Visual detection results obtained with the YOLOv8 model

Figure 9 presents examples in which the system, using the YOLOv8 model, classifies the scene as SUSPECT because it detects multiple elements that conceal the user's face, such as masks, sunglasses, or hats. In both cases, the model's bounding boxes indicate the identified classes and their corresponding confidence levels in a neutral ATM environment.

Fig. 9. Visualization of the 'SUSPECT' state generated by the YOLOv8-based analysis system

Figure 10 presents an example in which the system, using the YOLOv8 model, classifies the scene as NORMAL. In this case, the user's face is fully visible and shows no elements that conceal it in a neutral ATM environment.

Fig. 10. Visualization of the 'NORMAL' state generated by the YOLOv8-based analysis system

Table 2 presents the performance of the YOLOv8 model for each evaluated class. The results show high precision across all categories, indicating that the model tends to generate correct predictions with few false positives. However, recall is moderate, showing that although the model is accurate during detection, it may miss some instances present in the validation dataset. The hats class achieved the highest precision (1.000), followed by sunglasses and masks, while the mAP@0.5 values remained stable between 0.692 and 0.772, reflecting good overall performance. The behavior of mAP@0.5:0.95 shows the expected decrease due to stricter IoU thresholds, although the model maintains competitive performance in scenarios with visual variability and partial facial occlusions.

Tab. 2. Class-wise results – YOLOv8 (Precision, Recall, and mAP)

Class	Instances	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Masks	75	0.884	0.507	0.692	0.378
Hats	39	1.000	0.538	0.769	0.457
Sunglasses	29	0.944	0.586	0.772	0.432

Table 3 summarizes the overall performance of the YOLOv8 model used in this research. The model was implemented using Ultralytics version 8.3.228, with PyTorch 2.8.0 as the main framework. The evaluation was performed on a validation dataset of 57 images, containing 143 annotated instances across the classes masks, hats, and sunglasses.

The average inference time was 201.3 ms per image, measured under the experimental configuration used in this study. This result should be interpreted with the understanding that the model was executed on a CPU, which directly affects processing speed. Therefore, the reported time does not represent an absolute measure of the YOLOv8's potential but rather the performance observed under the study's execution conditions.

Tab. 3. Overall performance of the YOLOv8 model

Category	Description
Model	YOLOv8 (Ultralytics 8.3.228)
Framework	PyTorch 2.8.0
Validation images	57
Total instances	143
Inference time	201.3 ms per image

9. FASTER R-CNN TRAINING AND RESULTS

To train the Faster R-CNN model, a dataset annotated in PASCAL VOC format was prepared, following the same workflow used for YOLO: loading, verification, training, and validation split, and class distribution analysis. This dataset enabled the model to learn to detect masks, sunglasses, and hats, which served as visual indicators in the "NORMAL" or "SUSPECT" classification system.

Figure 11 summarizes the dataset's characteristics. The upper bar chart shows the number of instances per class, with masks more prevalent than sunglasses. This imbalance later appeared in the model metrics. The bounding box overlap graph indicates that objects are primarily concentrated in the central region, consistent with the typical face position in the images. The coordinate and dimension heatmaps show consistent patterns in object width, height, and location, confirming that the dataset contains sufficient variability for training.

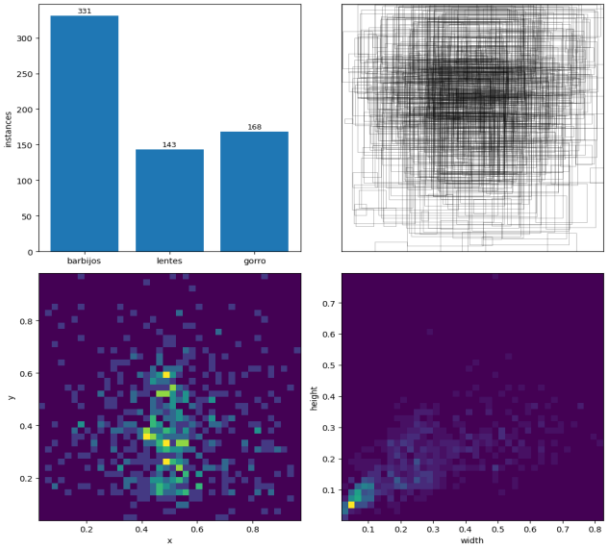

Fig. 11. Dataset used for Faster R-CNN training

Figure 12 shows the outputs from the Faster R-CNN model after inference on different images. The results display the bounding boxes and labels assigned by the detector for the classes masks, sunglasses, and hats. The predictions indicate that the model can identify multiple instances per image while maintaining consistent localization across faces despite variations in pose, lighting, and distance.

The detections confirm the characteristic behavior of Faster R-CNN: good spatial localization in controlled scenarios and reduced precision in scenes with high crowd density, occlusions, or low visual quality. Greater stability is also observed for the masks class, whereas sunglasses and hats show higher confidence variation, consistent with the metrics obtained during quantitative validation.

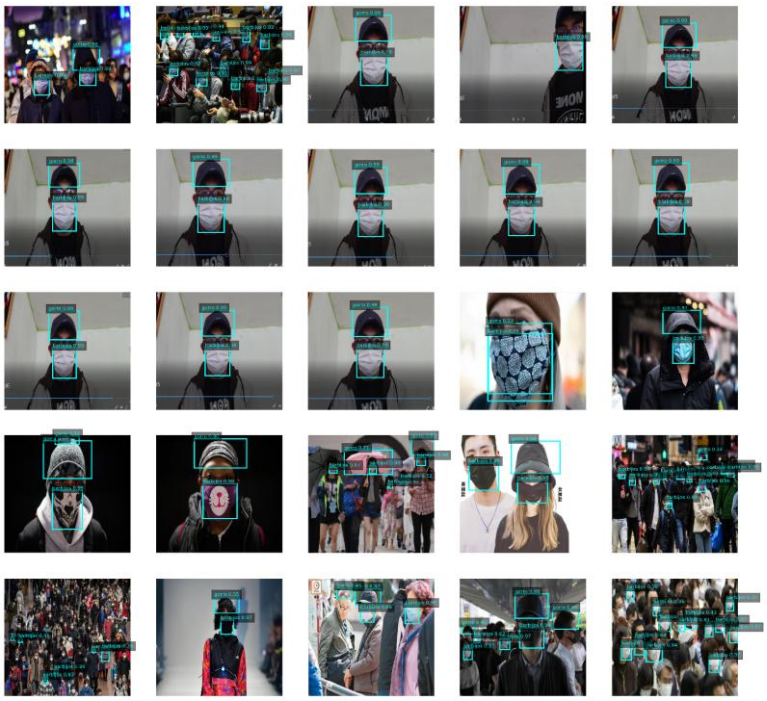

Fig. 12. Visual detection results obtained with Faster R-CNN

Figure 13 presents examples in which the system classifies the person as SUSPECT based on detections from the Faster R-CNN model during inference. In these cases, the model simultaneously identifies multiple elements that reduce facial visibility, such as sunglasses, masks, and hats, and assigns high confidence scores (above 0.90 in some cases).

The system integrates these detections using decision logic that considers the combined presence of objects covering critical facial regions. When this combination matches patterns associated with facial concealment, the final classifier automatically assigns the SUSPECT state. The images show the model-generated bounding boxes, detected classes, and confidence levels, allowing visual verification of how detecting multiple concealment elements leads to a transition toward an alert condition.

Fig. 13. Visualization of the 'SUSPECT' state generated by the Faster R-CNN-based analysis system

Figure 14 shows a case in which the Faster R-CNN model classifies the scene as NORMAL because it does not detect accessories associated with critical facial concealment. In this case, the system detects only a hat and does not detect sunglasses or masks, which are defined as risk indicators. The model analyzes the generated bounding boxes and their confidence scores. Because no combinations match the "suspicious" pattern, the decision logic determines that the face remains mostly visible, assigning the final NORMAL state.

Fig. 14. Visualization of the 'NORMAL' state generated by the Faster R-CNN-based analysis system

Table 4 summarizes the performance of the Faster R-CNN model for each evaluated class. The results show higher recall across all categories, indicating that the model identifies most instances in the validation dataset. However, precision is lower, indicating a tendency to generate additional predictions or less accurate detections.

The masks class achieved the best overall performance, reaching an mAP@0.5 of 0.807, whereas hats and sunglasses showed lower metrics due to the visual variability of these objects and the smaller number of available instances. The mAP@0.5:0.95 value shows the expected decrease, since this criterion applies stricter IoU thresholds.

Tab. 4. Class-wise Faster R-CNN results – (Precision, Recall, and mAP)

Class	Instances	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Masks	83	0.411	0.892	0.807	0.379
Hats	32	0.176	0.781	0.661	0.304
Sunglasses	26	0.149	0.654	0.560	0.184

Table 5 summarizes the overall performance of the trained model. The Faster R-CNN architecture with a ResNet50-FPN backbone was implemented in PyTorch and evaluated on a validation dataset of 57 images, containing 141 annotated instances across the classes masks, hats, and sunglasses. The average inference time was 103.3 ms per image.

Although Faster R-CNN is a two-stage architecture, in this study, it achieved a lower inference time than YOLOv8 because it was executed on a GPU. Therefore, this difference should be attributed mainly to the hardware conditions used, not to a general superiority of Faster R-CNN in terms of speed.

Tab. 5. Overall performance of the Faster R-CNN model

Category	Description
Model	Faster R-CNN ResNet50-FPN
Framework	PyTorch
Validation images	57
Total instances	141
Inference time	103.3 ms per image

10.CLASS-WISE PERFORMANCE COMPARISON BETWEEN YOLOV8 AND FASTER R-CNN

This section presents a comparative analysis of the results from the YOLOv8 and Faster R-CNN models, using standard object detection evaluation metrics: Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95. Table 6 summarizes the performance of both models for each evaluated class (masks, hats, and sunglasses), enabling the identification of differences in localization capability, sensitivity, and generalization.

Tab. 6. Comparative class-wise results – YOLOv8 vs Faster R-CNN

Class	Metric	YOLOv8 (Tab. 2)	Faster R-CNN (Tab. 4)
Masks	Images	43	43
	Instances	75	83
	Precision (P)	0.884	0.411
	Recall (R)	0.507	0.892
	mAP@0.5	0.692	0.807
	mAP@0.5:0.95	0.378	0.379
Hats	Images	32	32
	Instances	39	32
	Precision (P)	1.000	0.176
	Recall (R)	0.538	0.781
	mAP@0.5	0.769	0.661
	mAP@0.5:0.95	0.457	0.304
Sunglasses	Images	27	27
	Instances	29	26
	Precision (P)	0.944	0.149
	Recall (R)	0.586	0.654
	mAP@0.5	0.772	0.560
	mAP@0.5:0.95	0.432	0.184

Table 6 shows that YOLOv8 achieved higher precision across the three evaluated classes: 0.884 for masks, 1.000 for hats, and 0.944 for sunglasses, compared with 0.411, 0.176, and 0.149 for Faster R-CNN. This indicates that YOLOv8 produces fewer false positives and therefore reduces the likelihood of false alarms in ATM environments. By contrast, Faster R-CNN achieved higher recall across all classes: 0.892 for masks, 0.781 for hats, and 0.654 for sunglasses, compared with 0.507, 0.538, and 0.586 for YOLOv8. This demonstrates that Faster R-CNN detects more objects in the scene, reducing the risk of false negatives. In ATM security applications, this is particularly relevant because a false negative may mean failing to detect a potential facial concealment condition.

In real-world scenarios, Faster R-CNN would be more suitable if the system's priority is to avoid missing suspicious situations, since its higher recall enables the detection of more possible concealed events. However, if the objective is to reduce unnecessary alerts and avoid frequent false alarms, YOLOv8 would be the more stable option. For operational deployment, YOLOv8 could serve as the primary detector, while Faster R-CNN could serve as a support model in higher-risk scenarios.

11.CONCLUSIONS

The analysis shows that the applied detection models can automate the detection of elements associated with suspicious behavior in ATM environments. Both YOLOv8 and Faster R-CNN were evaluated to determine their effectiveness under real-world conditions, accounting for variations in lighting, partial occlusions, accessory use, and the number of people in the scene.

The comparative analysis between YOLOv8 and Faster R-CNN shows that each model exhibits distinct detection characteristics, reflecting their architectural designs. YOLOv8 maintains a balance between precision and speed, enabling real-time operation, whereas Faster R-CNN achieves higher recall and stricter mAP values, allowing detection of a greater number of objects, even under reduced visibility or facial occlusion.

For the masks class, YOLOv8 achieved a precision of 0.884 compared to 0.411 for Faster R-CNN, indicating that YOLOv8 generates fewer false positives. However, Faster R-CNN achieved a recall of 0.892 compared to 0.507 by YOLOv8, demonstrating greater capability to identify real instances, as reflected in the mAP@0.5 metric, where Faster R-CNN reached 0.807 compared to 0.692 by YOLOv8. In the hats class, YOLOv8 achieved a precision of 1.000, whereas Faster R-CNN obtained 0.176, demonstrating a substantial difference in favor of YOLOv8 for detection precision. Nevertheless, Faster R-CNN achieved a recall of 0.781, surpassing the 0.538 obtained by YOLOv8 and showing better performance across variations in object shape and position. For the sunglasses class, YOLOv8 achieved a precision of 0.944, whereas Faster R-CNN reached 0.149. Despite this, Faster R-CNN achieved a recall of 0.654, compared to 0.586 for YOLOv8, once again indicating a greater ability to recover existing objects within the scene, as reflected in the mAP@0.5 and mAP@0.5:0.95 values.

Although the obtained results are satisfactory, the study presents certain limitations, mainly related to the dataset's size and imbalance, which could affect model generalization for less-represented classes such as sunglasses. In addition, the experiments were conducted with a single type of pinhole camera under controlled conditions, limiting the system's extrapolation to more diverse operational environments. As future work, it is proposed to expand and diversify the dataset, evaluate more recent transformer-based architectures, optimize the models for edge-device deployment, and explore probabilistic classification mechanisms to complement the current deterministic logic. Finally, pilot testing in real ATM environments is recommended to validate system performance under authentic operational scenarios.

Conflicts of Interest

The authors declare no conflict of interest.

REFERENCES

Amirgaliyev, B., Mussabek, M., Rakhimzhanova, T., & Zhumadillayeva, A. (2025). A review of machine learning and deep learning methods for person detection, tracking and identification, and face recognition with applications. *Sensors, 25*(5), Article 1410. https://doi.org/10.3390/s25051410

Duong, H.-T., Le, V.-T., & Hoang, V. T. (2023). Deep learning-based anomaly detection in video surveillance: A survey. *Sensors, 23*(11), Article 5024. https://doi.org/10.3390/s23115024

Elrahman, M. A., Elbahri, F., & Zhao, C. (2025). Deep BiLSTM attention model for spatial and temporal anomaly detection in video surveillance. *Sensors, 25*(1), Article 251. https://doi.org/10.3390/s25010251

Feng, Y., Yu, S., Peng, H., Li, Y.-R., & Zhang, J. (2022). Detect faces efficiently: A survey and evaluations. *IEEE Transactions on Biometrics, Behavior, and Identity Science, 4*(1), 1–18. https://doi.org/10.1109/TBIOM.2021.3120412

Hermens, F. (2024). Automatic object detection for behavioural research using YOLOv8. *Behavior Research Methods, 56*(7), 7307–7330. https://doi.org/10.3758/s13428-024-02420-5

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines, 11*(7), Article 677. https://doi.org/10.3390/machines11070677

Ihsan, U., Jhanjhi, N. Z., Ashraf, H., Ashfaq, F., & Wicaksana, F. A. (2025). A real-time intelligent surveillance system for suspicious behavior and facial emotion analysis using YOLOv8 and DeepFace. *Engineering Proceedings, 59*, Article 59. https://doi.org/10.3390/engproc2025107059

Ingle, P. Y., & Kim, Y.-G. (2022). Real-time abnormal object detection for video surveillance in smart cities. *Sensors, 22*(10), Article 3862. https://doi.org/10.3390/s22103862

Ju, R.-Y., & Cai, W. (2023). Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. *Scientific Reports, 13*, Article 20077. https://doi.org/10.1038/s41598-023-47460-7

Khalili, B., & Smyth, A. W. (2024). SOD-YOLOv8-Enhancing YOLOv8 for small object detection in aerial imagery and traffic scenes. *Sensors, 24*(19), Article 6209. https://doi.org/10.3390/s24196209

Kim, J., & Cho, J. (2021). RGDiNet: Efficient onboard object detection with Faster R-CNN for air-to-ground surveillance. *Sensors, 21*(5), Article 1677. https://doi.org/10.3390/s21051677

Li, X., Hao, T., Li, F., Zhao, L., & Wang, Z. (2023). Faster R-CNN-LSTM construction site unsafe behavior recognition model. *Applied Sciences, 13*(19), Article 10700. https://doi.org/10.3390/app131910700

Makhlouf, A., Ben Ali, M., & Al-Ali, A. (2024). Advances in computer vision and deep learning and its applications. *Electronics, 14*(8), Article 1551. https://doi.org/10.3390/electronics14081551

Mittal, P. (2024). A comprehensive survey of deep learning-based lightweight object detection models for edge devices. *Artificial Intelligence Review, 57*(9). https://doi.org/10.1007/s10462-024-10877-1

Rahim, A., Zhong, Y., Ahmad, T., Ahmad, S., Pławiak, P., & Hammad, M. (2023). Enhancing smart home security: Anomaly detection and face recognition in smart home IoT devices using logit-boosted CNN models. *Sensors, 23*(15), Article 6979. https://doi.org/10.3390/s23156979

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). IEEE. https://doi.org/10.1109/CVPR.2016.91

Selvi, E., Adimoolam, M., Karthi, G., Thinakaran, K., Balamurugan, N. M., Kannadasan, R., Wechtaisong, C., & Khan, A. A. (2022). Suspicious actions detection system using enhanced CNN and surveillance video. *Electronics, 11*(24), Article 4210. https://doi.org/10.3390/electronics11244210

Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction, 5*(4), 1680–1716. https://doi.org/10.3390/make5040083

Thaer, T., Majdi, M., Muhammed, S., Hakim, A., & El-Saleh, A. A. (2025). A comprehensive review of face detection techniques for occluded faces: Methods, datasets, and open challenges. *Computer Modeling in Engineering & Sciences, 143*(3), 2615–2673. https://doi.org/10.32604/cmes.2025.064857

Trigka, M., & Dritsas, E. (2025). A comprehensive survey of machine learning techniques and models for object detection. *Sensors, 25*(1), Article 214. https://doi.org/10.3390/s25010214

Wu, H., Zheng, Z., Lv, L., Xu, Y., Bardou, D., Niu, S., Yu, G., & Wang, Y. (2025). A spatially aware global and local perspective approach for few-shot incremental learning. *Scientific Reports, 15*(1), Article 8323. https://doi.org/10.1038/s41598-025-08323-5

Wu, W., Yin, Y., Wang, X., & Xu, D. (2019). Face detection with different scales based on Faster R-CNN. *IEEE Transactions on Cybernetics, 50*(10), 1–12. https://doi.org/10.1109/TCYB.2018.2859482

Xiao, Y., Wang, X., Zhang, P., Meng, F., & Shao, F. (2020). Object detection based on Faster R-CNN algorithm with skip pooling and fusion of contextual information. *Sensors, 20*(19), Article 5490. https://doi.org/10.3390/s20195490

Yaseen, M. (2024). *What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector*. arXiv. https://doi.org/10.48550/arXiv.2408.15857

Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). *FaceBoxes: A CPU real-time face detector with high accuracy*. arXiv. https://doi.org/10.48550/arXiv.1708.05234

Zhang, W., & Gu, X. (2023). Few shot class incremental learning via efficient prototype replay and calibration. *Entropy, 25*(5), Article 776. https://doi.org/10.3390/e25050776