EURASIAN JOURNAL OF TECHNOLOGY AND INNOVATION
Innovative Academy Research Support Center
IF= 9.1 www.in-academy.uz/index.php/ejti

IMPROVING THREAT DETECTION EFFICIENCY IN INTELLIGENT SECURITY SYSTEMS BASED ON EDGE ARTIFICIAL INTELLIGENCE TECHNOLOGIES

Bozorov Abdimannon Abduroimovich
Public Safety University of the Republic of Uzbekistan
Lecturer of the Department
Tashmanov Yerjan Baymatovich
Public Safety University of the Republic of Uzbekistan
Head of the Department, Doctor of Technical Sciences, Professor
https://doi.org/10.5281/zenodo.21060193

ARTICLE INFO

Received: 02nd June 2026
Accepted: 08th June 2026
Online: 09th June 2026

KEYWORDS

Intelligent security system, video surveillance, Edge Artificial Intelligence, threat detection, local computing device, network traffic, latency, central server load, risk-level assessment, continuous monitoring.

ABSTRACT

This article investigates the problem of improving threat detection efficiency in intelligent security systems through the use of Edge Artificial Intelligence technologies. It is substantiated that the transmission of complete video surveillance streams to a central server increases network traffic, latency, and computational load. In this regard, the article proposes a multi-stage intelligent detection model based on performing video preprocessing, object detection, object tracking, and risk-level assessment directly on a surveillance camera or a local computing device. By transmitting only high-risk episodes, key frames, and metadata to the central server, the model reduces network load, shortens processing time, and supports continuous real-time security monitoring. The study develops mathematical expressions for evaluating threat probability, detection accuracy, processing time, network load, and overall system efficiency. The obtained results demonstrate that the functional distribution of edge and central computing capabilities increases the overall effectiveness of the system while maintaining the quality of threat detection.

INTRODUCTION

Over the last decade, video surveillance systems have been transforming from tools that merely store archival records into intelligent monitoring environments capable of detecting hazardous events in real time. This transformation has been driven, on the one hand, by the decreasing cost of cameras and, on the other hand, by advances in computer vision and neural networks. In practice, however, many multi-facility security systems still operate on the basis of centralized servers: camera streams are transmitted to a data center, the analysis is performed there, and the result is subsequently returned to the operator. The main drawback of this approach is that it imposes high requirements on network bandwidth, latency, privacy, and computational resources. Edge-

based solutions mitigate precisely this contradiction: data are processed close to the source, only the necessary metadata or episodes are transmitted to the center, and, as a result, response time is improved and traffic is reduced. [3]

The advantage of edge analytics is particularly evident in data-intensive domains such as video surveillance. For video analytics to be useful, the system must respond within a very short time while an event is occurring. However, the continuous delivery of high-resolution streams to a central facility in both offline and online modes, especially in facilities with multiple cameras, can rapidly saturate even broadband networks. The FilterForward system developed for constrained edge nodes addresses this exact problem by proposing a semantic filtering approach in which only 'interesting' frames from thousands of cameras are transmitted. The authors demonstrated that such a solution mitigates the network bottleneck and delivers more useful frames in real urban-camera environments. The key idea of that study is that dangerous events occur relatively rarely; therefore, transmitting the entire raw stream is not necessary. [4]

Recent scientific reviews show that individual components of this field have advanced considerably, whereas integrated, practical, multi-stage architectures have not yet been sufficiently standardized. A comprehensive 2023 review examined more than 30 streaming video analytics systems according to 17 criteria and emphasized that, although many studies improve separate components, issues related to network management, control, adaptability, privacy, and resilience across the full system pipeline remain insufficiently addressed. A 2025 review on the deployment of edge artificial intelligence proposed a five-stage approach for managing the multi-criteria balance between lightweight architectures, quantization, pruning, and hardware adaptation. Thus, the scientific problem has shifted from improving a single model to optimizing the stream architecture of the entire surveillance system. [5]

When selecting models for video surveillance, speed and compactness are of particular importance. In the field of rapid object detection, the YOLOv7 family has demonstrated a combination of high speed and high accuracy for real-time operation; for mobile and resource-constrained environments, hardware-adapted lightweight architectures such as MobileNetV3 have been developed. In continuous object tracking, the DeepSORT algorithm, which relies on a deep association criterion, is noteworthy because it reduces identity switching under occlusion and reappearance conditions. Therefore, a modern intelligent security system is usually built around the chain of 'detection–tracking–risk assessment', and moving this chain to the local level is becoming the main practical objective of edge artificial intelligence. [6]

Reliable datasets that provide experimental grounds for this area have also been formed. The UCF-Crime dataset contains 1,900 long, untrimmed surveillance videos totaling 128 hours and covers 13 real anomaly classes; XD-Violence contains 4,754 videos totaling 217 hours and enables the assessment of

violence-related cases in a multimodal context. These datasets show that threats are typically short, context-dependent, and often complicated by illumination and background changes. Consequently, local preprocessing, context retention, and risk-level assessment are not merely issues of reducing network usage, but also issues of preserving detection quality. [7]

In this regard, the present article substantiates a new multi-stage model that eliminates the shortcomings of a centralized architecture while preventing the loss of accuracy associated with purely local analysis. The central idea of the model is as follows: a camera or local device first prepares the frame, detects objects, evaluates movement and zone violations, and computes the threat probability; the central server then performs deeper verification, archiving, and system-level management only for episodes that exceed the risk threshold. From both theoretical and experimental perspectives reported in the literature, this approach appears to be the most promising solution. [8]

Problem Statement

As the number of surveillance cameras at modern protected facilities increases, centralized processing architectures aggravate four interrelated problems. First, transmitting all streams to the center consumes the main network resource. Second, because analysis is performed centrally, the signal path becomes longer and the delay between an event and an alert increases. Third, object detection, tracking, behavior analysis, archiving, and search tasks are concentrated on the central server, creating high pressure on computing and memory resources. Fourth, the transmission of all raw video over the network creates additional risks in terms of privacy and information security. The literature notes that large-scale video streams represent a ‘heavy’ workload for video analytics in terms of both network bandwidth and latency, and this problem is particularly pronounced in multi-camera practice. [9]

The second aspect of the problem is that it is not sufficient simply to ‘move’ a centralized system to the camera. Although a purely local approach imposes the lowest requirements on traffic, accuracy may decrease in situations involving complex behavior, low-light environments, small objects, and multi-camera context. EVA experiments show that the camera-only mode has the lowest latency but lags behind server-based and edge–server collaborative modes in terms of accuracy. A separate study on low-light conditions demonstrated that, when image quality deteriorates, a conventional object-detection stage increases the number of false positives; therefore, an image-enhancement stage must also be included in the local processing chain. [10]

Accordingly, the scientific and practical task can be formulated as follows: it is necessary to develop a multi-stage intelligent model for video surveillance and technical security tools that enables data processing at the facility itself, real-time threat detection, reduction of network traffic, reduction of central server load, and continuity of monitoring. The model must simultaneously satisfy two opposing

requirements: the local stage should provide high speed and high recall, whereas the central stage should provide high accuracy and additional contextual verification. This idea was experimentally confirmed in the work of Song and Nang: the edge stage provides high recall, while the server stage provides high precision; when integrated, false alarms decrease and the probability of missing important episodes also declines. [11]

Thus, the main scientific question within the article is formulated as follows: how can overall system efficiency be improved by performing object detection, tracking, and risk-level assessment on a local device and transmitting to the center only those segments with a sufficient threat probability? To answer this question, architecture, algorithm, mathematical expressions, and result-evaluation criteria must be combined within a single conceptual model. [12]

The practical objectives of this study are defined as follows: to substantiate the feasibility of local preprocessing of data in the video surveillance stream; to determine criteria for selecting a lightweight object-detection model and tracking modules; to develop a local threat-assessment mechanism; to shift network transmission rules from a 'full stream' approach to a 'selective transmission' approach; and finally, to calculate the integral balance among accuracy, time, network load, and server load. [13]

Analysis of Related Scientific Research

The literature on edge video analytics can be divided into two broad groups: the first group studies systems that optimize streams from the standpoint of transport and resources, whereas the second group examines systems that semantically analyze dangerous events. The 2023 review conducted by Ravindran analyzed more than 30 systems according to 17 criteria and showed that existing studies do not address network management, scheduling, adaptive control, security, privacy, fault tolerance, and experimental foundations at the same level. An important conclusion of the review is that accelerating a separate algorithm alone is insufficient; the entire sequential processing chain among the camera, edge node, and server must be redesigned. [14]

From the perspective of architecture selection, the FilterForward and EVA studies represented an important turning point. FilterForward, based on the principle of semantic filtering, sends only interesting events to the center and reduces computational cost by sharing common features on a multi-application edge node. EVA treats the region where the object is located as a small transmission unit and suppresses spatial and temporal redundancies. As a result, traffic is reduced not by transmitting less indiscriminately, but by preserving useful information. These approaches are highly important for security systems because they enable the analysis of dangerous events not over the entire frame, but specifically over the threat source and its nearby context. [15]

In selecting object-detection models, the balance between real-time performance and accuracy remains the key criterion. YOLOv7 is notable for demonstrating high speed and high accuracy on powerful graphics

processors, while its lightweight variants are suitable for edge devices. MobileNetV3 is regarded as a convenient backbone network for mobile and embedded devices due to its hardware-aware search and compact architecture. In inter-object association and trajectory maintenance, DeepSORT and related approaches are useful in scenes involving occlusion, temporary disappearance from frames, and reappearance. This provides more stable results than simple frame-by-frame classification in security tasks such as detecting 'zone violations', 'prohibited movement directions', 'abandoned objects', 'fights', and 'falls'. [16]

For practical operation on edge devices, neural network compression is becoming a mandatory requirement. A 2025 review on edge intelligence identifies model pruning, quantization, knowledge distillation, and hardware-aware neural architecture search as the main supporting techniques. The authors note that the principal obstacles to deploying high-accuracy models on edge devices are the number of parameters, memory, and energy constraints; in particular, they state that INT8 quantization can reduce model size by up to 75% and accelerate inference without a significant loss of accuracy. Another 2025 review shows that reaching an optimum requires selecting not only the model but also the hardware on which it is deployed, giving as an example a balanced point on an edge neural processor with 132 milliseconds of latency, 91.5% accuracy, and 4.1 watts of power consumption. [17]

The system proposed by Maltezos et al. for person detection and situational monitoring on edge devices is also noteworthy. It provides for real-time person detection, key-frame identification, and direct transmission of geolocated signals from the edge device. The authors show that when computing, storage, and information-exchange processes are integrated into a single device, the system becomes fast and flexible. Barthélemy et al. proposed an edge-sensor concept for urban traffic monitoring that does not transmit raw images, but sends only indicators and metadata; they justify the transmission of denatured results rather than images themselves specifically for the purposes of privacy protection and traffic reduction. These approaches are also fully applicable to protected facilities. [18]

The 2024 study by Ali et al. on detecting abnormal events directly on edge nodes is of particular importance. In that work, a deep model was deployed on edge nodes to distinguish 13 types of unusual events. The experimental results showed that the proposed model achieved a test accuracy of 80.92%; its quantized version on a Raspberry Pi 4 B device achieved 32.21 frames per second while almost preserving accuracy at 79.90%, whereas the non-quantized model operated much more slowly at 12.14 frames per second, and energy efficiency improved by 2.77 times. These results indicate that, on an edge node, 'sufficient accuracy for operational performance and high speed' is often more important than 'absolute maximum accuracy'. [19]

Adaptation to low-light environments is not sufficiently considered in many practical systems. Abu Awwad et al.

placed an image-enhancement model before the conventional detection stage in a smart camera system operating under low-light conditions and showed that object-detection accuracy improved by more than 20%. The authors reported that the local classifier distinguished bright and low-light scenes with 85.24% accuracy, while the entire chain produced a result in approximately one second. This means that, in a security system, edge artificial intelligence should not consist only of a detection model; it should also include image quality, environmental properties, and preparation suited to the subsequent stage. [20]

Song and Nang demonstrate the practical superiority of an edge–server hybrid approach in a large-scale indoor video surveillance environment with concrete numerical evidence. In their experiment, for 20 video streams, the edge-only architecture processed the data in 294 seconds and produced an F1 value of 0.848 for the assault class, while the server-only architecture required 1040 seconds and produced F1=0.889. The integrated edge–server architecture produced F1=0.904 in 314 seconds. For the fall class, the corresponding values were 0.918, 0.923, and 0.942. Thus, the hybrid structure operated approximately 3.31 times faster than the server-only mode, while also outperforming it in terms of the F1 indicator. In the separate day-and-night analysis, the average F1 value of the integrated architecture was 0.926, compared with 0.883 for edge-only and 0.912 for server-only; accordingly, the integrated method showed approximately 4.9% improvement over edge-only and about 1.6% improvement over server-only. [21]

Along with the above studies, an important direction has also emerged in resource management. The CogVSM model of Ugli et al. achieved up to 32.1% memory savings in a multi-tier edge computing system by intelligently freeing GPU memory and managing model reloading during intervals when objects occur infrequently. The authors also showed that the approximately three-second delay associated with model reloading was prevented through EWMA-guided prediction. This result means that, for a central server or a higher-level edge node, not only the inference model itself but also the model-loading policy forms part of overall efficiency. [22]

Based on the above literature analysis, the following conclusion can be drawn: existing studies have thoroughly addressed certain components of traffic reduction, accuracy improvement, and efficient resource allocation; however, a unified functional model based specifically on the principle of ‘preprocessing on the camera or local device + object detection + risk-level assessment + selective transmission’ for technical security tools has not been sufficiently formulated. The scientific novelty of the present article is directed at filling this gap. [23]

Table 1

Practical results of selected studies on edge video analytics.

Research area	Architecture type	Main experimental result	Significance for the security system

Semantic filtering and semantic transmission	Edge–center	Only interesting frames are transmitted, reducing the network bottleneck	Dangerous episodes can be transmitted instead of raw video
Suppression of spatial-temporal redundancy	Edge–server	Traffic is reduced by up to 90% while high accuracy is maintained	Suitable for facilities with limited network traffic capacity
Local detection of abnormal activity	Edge-only	Quantization increases speed and energy efficiency	Real-time operation is ensured on small devices
Low-light image enhancement	Edge-only	Accuracy improves by more than 20%	Important for night surveillance and warehouse areas
Hybrid management of large-scale streams	Edge–server	High F1 and rapid processing are achieved for 20 streams	Central server load decreases
Hierarchical memory management	Hierarchical edge	GPU memory consumption decreases by up to 32.1%	Deployment of additional artificial intelligence services becomes easier

The generalized results in the table are presented on the basis of the works of FilterForward, EVA, Ali et al., Abu Awwad et al., Song and Nang, and Ugli et al. The table shows that the real advantage is observed precisely in systems that are multi-stage, based on selective transmission, and functionally separate the edge and server components. [24]

Research Methodology

This study was carried out on the basis of a systems approach. In the first stage, major scientific sources from 2018 to 2026 on video surveillance, edge computing, anomaly detection, and security systems were analyzed. The aim of this stage was to identify the technical roots of the problem, namely which tasks can be efficiently performed locally and which tasks should remain at the center. In the second stage, architectural synthesis was conducted and the functional distribution among the camera, local device, and central server was developed. In the third stage, the mathematical apparatus of the model was constructed. In the fourth stage, meta-analytical verification was performed on the basis of experimental results published in the literature. [25]

The conceptual basis of the methodology relied on the principle of ‘near-source analysis – selective transmission – deep verification at the center’. In this approach, the camera or the local computing device connected to it first cleans the frame, evaluates illumination conditions, detects relevant objects, tracks them over a short time interval, and computes the threat probability. Only when the risk value exceeds the established threshold or the confidence of the local model decreases are key frames, a short video fragment, and metadata transmitted to the central server. This methodology reduces unnecessary network load while transforming the role of the central node from a ‘center that computes everything’

into a 'center that verifies and aggregates the analysis'. [26]

Five main indicators were selected as evaluation criteria: threat probability, detection accuracy, processing time, network load, and overall system efficiency. Detection quality is evaluated not only by simple accuracy but also by precision, recall, and F1 indicators, because the costs of false positive and false negative results are not the same for security systems. Processing time is considered as a stage-by-stage sum, while network load is examined as a normalized indicator relative to the fully transmitted stream. Overall system efficiency is expressed by reducing several criteria to a single integral indicator. This approach is consistent with the nature of edge solutions, since they always manage the balance among 'accuracy–latency–traffic–resources'. [27]

Several reliable reference sources were selected for evaluating the experimental results. The UCF-Crime and XD-Violence datasets were used as references for general recognition of dangerous events; Song and Nang's 20-stream indoor video surveillance experiment was used for large-scale streams; EVA experiments were used for network load; Ali et al.'s Raspberry Pi experiment was used for speed and energy on a resource-constrained device; Abu Awwad et al.'s results were used for low-light environments; and CogVSM values were used for server load. Importantly, although these results were obtained for different tasks, they can be brought into a single integral analysis by normalizing them to the [0;1] interval. [28]

In this methodology, the experimental part uses two forms. The first is 'primary literary experimentation', in which measured results from sources are presented directly. The second is 'meta-analytical integration', in which indicators from different sources are normalized on the basis of a unified mathematical model and the overall benefit of the proposed architecture is evaluated. This method cannot fully replace a real laboratory test conducted in one place; however, it provides a sufficient analytical basis for demonstrating the architectural advantage with high confidence. [29]

Proposed Model

The proposed model transfers video surveillance systems from centralized analysis to multi-stage edge analysis. Its main idea is that, instead of transmitting raw information from the video stream entirely to the center, the information should be divided into useful semantic features and processed step by step. In this case, the camera or the local device located near the camera first performs the role of a 'sensing and filtering' unit, while the central server shifts to the role of a 'verifying, coordinating, and archiving' unit. As a result, the continuous inference load on the central server is sharply reduced, the amount of raw data transmitted through the network decreases, and response time is shortened. [30]

The first stage is the local preparation and frame-filtering stage. At this stage, frame quality, illumination conditions, the presence of movement, and the useful area are determined. The aim is to prevent static, safe, and low-information frames from being passed to subsequent

complex analysis. Under low-light conditions, the image-enhancement model is activated; in bright scenes, this stage is skipped. This approach reduces the number of incorrect object detections in low-light environments and provides high-quality input data for the subsequent stages. [20]

The second stage is the object-detection and tracking stage. At this stage, a lightweight detector suited to a resource-constrained environment is used. From a practical standpoint, a lightweight version of the YOLOv7 family or detectors based on MobileNetV3 are appropriate. Once an object is detected, it is tracked across a sequence of frames. The tracking module, for example algorithms from the DeepSORT, SORT, or ByteTrack classes, takes into account the object's position, direction of movement, speed, and trajectory over time. This is necessary for subsequently distinguishing 'normal movement' from 'dangerous movement'. [31]

The third stage is the formation of scene semantics and event features. Here, the information received from the detector and the tracking module is converted into risk indicators. These indicators include entry into a protected zone, movement along a prohibited trajectory, abrupt changes in speed, unexpected crowding, high-amplitude movements characteristic of a fight, falling, remaining motionless for a long time, an abandoned object, and the appearance of an unidentified object. The main achievement of this stage is that work with the video stream is elevated from the pixel level to the level of semantic features. As a result, the subsequent risk assessment answers not only the question 'what was seen?' but also the question 'what does it mean?' [32]

The fourth stage is the local risk-level assessment stage. This stage constitutes the main scientific novelty of the article. For each object and event, detection confidence, zone violation, movement anomaly, temporal stability, and scene context are reduced to a single risk indicator. The risk indicator is divided into three levels: low, medium, and high. In a low-risk situation, only a local log entry is generated. In a medium-risk situation, a key frame and brief metadata are sent to the center. In a high-risk situation, a 5–10-second video fragment, key frames, and a semantic summary are transmitted to the center as a package, and the local alert is also activated. Thus, only information that 'requires processing' reaches the server. [33]

The fifth stage is the central verification and multi-source fusion stage. The data received by the central server are reassessed using a deeper model. If an object has moved through important areas across several cameras, the central layer can perform object re-identification and reconstruct the sequence of events. However, this stage does not operate on a continuous raw stream; it works only on fragments and metadata selectively sent by the local stage. Therefore, central computing resources are used only for genuinely important scenes. [30]

The sixth stage is the model adaptation and long-term analysis stage. Over time, the central layer collects metadata received from local devices and learns which types of events occur more frequently in which zones. As a result,

risk thresholds, zone policies, and processing rules are readjusted according to the environment. When necessary, an updated lightweight model or updated threshold values are returned to the cameras. This transforms the system from a 'once-installed' structure into a structure that 'learns from its operating conditions'. [34]

The functional advantages of the proposed model can be explained as follows. First, traffic is reduced because selected episodes are transmitted instead of complete video. Second, latency is reduced because the initial decision is made locally. Third, central server load is reduced because the deep model operates not continuously but only on filtered streams. Fourth, the level of privacy increases because raw video does not continuously leave the local environment. Fifth, even when the local network is interrupted, a minimum level of monitoring is preserved, meaning that the system possesses continuity. [35]

Table 2

Functional structure of the proposed multi-stage intelligent detection model.

Stage	Task performed	Main output	Location
Local preparation	Assessment of frame quality, illumination, and motion; image enhancement if necessary	Cleaned and filtered frame	Camera or local device
Object detection	Detection of persons, vehicles, objects, and other monitored items	List of detected objects	Local device
Tracking	Monitoring trajectory, speed, entry/exit, and time-based status	Object trajectory and state parameters	Local device
Event features	Extraction of zone violation, falling, fighting, crowding, and abandonment features	Event vector	Local device
Risk assessment	Combining features to calculate the risk level	Low, medium, or high risk	Local device
Selective transmission	Sending only necessary frames, fragments, and metadata to the center	Signal package	Through the local network
Central verification	Verification with a deeper model and multi-camera fusion	Verified event and command	Central server
Long-term adaptation	Threshold readjustment and statistical adaptation	Updated policy and model	Central server

As this table shows, the proposed model frees the central server from the task of initial detection and transforms it into a high-value decision layer. This solution is most appropriate for technical security tools, especially in warehouses, production facilities, campuses, transport hubs, and protected facilities with limited network resources. [36]

Mathematical Model

The mathematical model of the proposed system is based on five key indicators: threat probability, detection quality, processing time, network load, and overall system efficiency. The

purpose of the model is not only to evaluate each indicator separately but also to integrate them into a unified decision-making logic. Here, reductions in time and traffic are assumed to be achieved not at the expense of accuracy but through multi-stage filtering. [27]

The threat probability $P_x(t)$ is determined as follows:

$$E = \lambda_1 Q_d + \lambda_2\left(1 - \hat{T}_q\right) + \lambda_3\left(1 - \hat{L}_t\right) + \lambda_4\left(1 - \hat{G}_s\right) + \lambda_5 U$$

where $s_o(t)$ is the object detection confidence, $s_h(t)$ is the behavioral anomaly feature, $s_z(t)$ is the protected-zone rule violation coefficient, $s_k(t)$ is the context coefficient, and α_i denotes the weights. If temporal stability must also be taken into account, the integral probability within a sliding window is obtained as follows:

$$\bar{P}_x(t) = 1 - \prod_{j=t-w+1}^{t} \left(1 - P_x(j)\right),$$

where w is the window length. This expression makes it possible to prioritize temporally stable threat states rather than short, one-time fluctuations. Low, medium, and high risk levels are defined respectively by $\bar{P}_x(t) < \tau_1$, $\tau_1 \le \bar{P}_x(t) < \tau_2$, and $\bar{P}_x(t) \ge \tau_2$.

Detection accuracy is evaluated using conventional classification criteria. Overall accuracy is defined as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN},$$

precision:

$$Pr = \frac{TP}{TP + FP},$$

recall:

$$Re = \frac{TP}{TP + FN},$$

and the F1 indicator:

$$F_1 = \frac{2Pr \cdot Re}{Pr + Re} = \frac{2TP}{2TP + FP + FN}.$$

In security systems, false negative results are often more dangerous than false positive results; therefore, the F_1 indicator is more informative in practical evaluation. If different weights need to be assigned to different threat classes, weighted accuracy is written as follows:

$$A_w = \frac{\sum_{i=1}^{N} \omega_i \cdot I(\hat{y}_i = y_i)}{\sum_{i=1}^{N} \omega_i},$$

where ω_i is the risk weight of the i-th event.

Processing time for the multi-stage model is determined as a sum:

$$T_q = T_t + T_a + T_k + T_h + T_b + T_u,$$

where T_t is preparation time (filtering, illumination assessment, image enhancement), T_a is detection time, T_k is tracking time, T_h is event-feature extraction time, T_b is risk-assessment time, and T_u is transmission time. In a centralized system:

$$T_q^m = T_u^{to'liq} + T_s,$$

where $T_u^{to'liq}$ is the time required to transmit the complete stream, and T_s is server inference time. In the proposed system, transmission time depends not on the complete stream but on selectively transmitted segments:

$$T_q^{ch} = T_t^{lok} + T_a^{lok} + T_k^{lok} + T_h^{lok} + T_b^{lok} + T_u^{meta}.$$

If $T_u^{meta} \ll T_u^{to'liq}$, the overall latency of the system is significantly reduced.

Network load is expressed as follows:

$$L_t = \frac{\sum_{i=1}^{N_c}\left(B_i^{meta} + \eta_i B_i^{parcha}\right)}{\sum_{i=1}^{N_c} B_i^{to'liq}},$$

where N_c is the number of cameras, B_i^{meta} is the metadata size for the i-th camera, B_i^{parcha} is the size of the video

fragment transmitted for a dangerous episode, η_i is the share of dangerous episodes for this camera, and $B_i^{to'liq}$ is the size of the complete stream. $L_t \in [0,1]$; as it approaches 1, the centralized architecture dominates, whereas as it approaches 0, the edge selective-transmission strategy becomes dominant.

Central server load is another important component of system efficiency. It can be represented as a normalized computing-load indicator as follows:

$$G_s = \frac{\sum_{j=1}^{M}(\mu_j C_j + \nu_j M_j)}{G_{max}},$$

where C_j is the computational cost of the j-th service, M_j is the memory requirement, μ_j and ν_j are their relative weights, and G_{max} is the maximum permissible combined server-load indicator. If non-dangerous streams are not sent to the center at all, the values of C_j and M_j are limited only to real dangerous episodes.

The following coefficient is introduced to assess monitoring continuity:

$$U = 1 - \frac{t_{uzilish}}{T_{obs}},$$

where $t_{uzilish}$ is the total time during which real-time analysis stops due to monitoring failure or network interruption, and T_{obs} is the total observation period. If the system can continue to operate independently at the local level, U remains high.

Based on the above, overall system efficiency is defined as follows:

$$E = \lambda_1 Q_d + \lambda_2(1 - \hat{T}_q) + \lambda_3(1 - \hat{L}_t) + \lambda_4(1 - \hat{G}_s) + \lambda_5 U,$$

where Q_d is the selected accuracy criterion (in practice, most often F_1), $\hat{T}_q$ is normalized processing time, $\hat{L}_t$ is normalized network load, $\hat{G}_s$ is normalized server load, $\lambda_i \geq 0$, $\sum\lambda_i = 1$. If speed and traffic are particularly important for the system designer, λ_2 and λ_3 are assigned larger values; if avoiding missed dangerous objects is the priority, λ_1 is increased. For the practical assessment in this article, $\lambda_1 = 0{,}35$, $\lambda_2 = 0{,}25$, $\lambda_3 = 0{,}25$, $\lambda_4 = 0{,}15$, and $\lambda_5 = 0$ are adopted. This choice is based on the assumption that, for a security system, accuracy is the priority, while time and traffic are also almost equally important.

The condition under which the proposed model is superior to the centralized architecture is written as follows:

$$E_{taklif} > E_{markaz}$$

or, in expanded form:

$$\lambda_1(Q_d^{taklif} - Q_d^{markaz}) + \lambda_2(\hat{T}_q^{markaz} - \hat{T}_q^{taklif}) + \lambda_3(\hat{L}_t^{markaz} - \hat{L}_t^{taklif}) + \lambda_4(\hat{G}_s^{markaz} - \hat{G}_s^{taklif}) > 0.$$

This expression shows that, even if the proposed system produces very small changes in accuracy under certain scenarios, gains in time, traffic, and server load can increase overall efficiency. The experiments reported in the literature confirm precisely this situation. [37]

Experiments and Results

The experimental part of the article relied on two sources: first, experimental results reported in the open scientific literature; and second, the reduction of these results to a normalized integral indicator based on the mathematical model described above. The purpose of this approach is to compare the general

architectural conclusions of different studies, although they were conducted for different tasks, devices, and datasets. [38]

The first reference result concerns network load and the efficiency of selective transmission. In the EVA system, the server-only mode on real video streams provided an average accuracy of 98.60% and normalized traffic of 0.97, whereas the edge-assisted mode suppressing spatial-temporal redundancy operated with only 0.08 normalized traffic while maintaining 97.10% accuracy. This indicates an approximate 91.8% reduction in normalized load compared with centralized transmission. On drone data, the proposed mode showed 94.76% accuracy with 0.25 normalized traffic. In the overall conclusion of the system, the authors noted that network load can be reduced by up to 90%. This shows that the mechanism of transmitting a 'suspicious region' instead of 'transmitting everything' has real practical value for protected facilities. [39]

The second reference result is the superiority of the hybrid architecture in large-scale streams. In Song and Nang's experiment, for the 'assault' class across 20 streams, the edge-only mode produced 294 seconds and F1=0.848, the server-only mode produced 1040 seconds and F1=0.889, and the integrated edge–server mode produced 314 seconds and F1=0.904. For the 'fall' class, the corresponding F1 values were 0.918, 0.923, and 0.942. Thus, the integrated solution operated much faster than the server-only mode and much more accurately than the edge-only mode. The calculation shows that, across 20 streams, the integrated method was approximately 3.31 times faster than server-only. [40]

Day-and-night evaluation is also an important indicator for security systems. In the separate results for assault and fall classes, the integrated architecture showed F1=0.902 for daytime assault, F1=0.906 for nighttime assault, F1=0.928 for daytime fall, and F1=0.968 for nighttime fall. The average of these four values is 0.926. The corresponding averages are 0.883 for edge-only and 0.912 for server-only. Therefore, the integrated solution achieved an average improvement of 4.9% over edge-only and approximately 1.6% over server-only. This is highly significant, because many systems make the most errors in real facilities precisely in evening and night conditions. [41]

The results of Ali et al. are important for assessing the efficiency of deploying a model on a local computing device. On a Raspberry Pi 4 B device, the quantized model preserved 79.90% accuracy and achieved 32.21 frames per second, whereas the non-quantized variant operated at 12.14 frames per second with 80.10% accuracy. Thus, although the difference in accuracy was only 0.20%, speed increased by more than 2.65 times. Energy efficiency was 3.88 frames per watt, which is 2.77 times higher than the 1.40 frames per watt of the non-quantized version. This confirms that, for the edge stage in a security system, it is necessary to choose a model adapted not for maximum accuracy alone but for fast and energy-efficient operation. [42]

Significant results have also been obtained in relation to illumination and image quality. Abu Awwad et al. applied a local image-enhancement stage before object detection and showed that detection quality improved by more than 20%. A local classifier designed to distinguish bright and low-light scenes achieved 85.24% accuracy in testing; individual chain components operated at the millisecond level, and the total response time across all stages was approximately one second. This result shows that, in video surveillance systems operating around the clock, the local preparation stage should be treated as an independent and separate design object. [20]

Central server memory load is also an important indicator of practical effectiveness. In the CogVSM experiment, predictive model management and targeted memory release reduced GPU memory consumption by up to 32.1%, while also preventing the approximately three-second model-reloading delay caused by the sudden appearance of objects. This result is particularly important for multi-service central servers that analyze video records in parallel: if one service consumes less memory, additional services can be deployed on a single server. [22]

The data of Song and Nang on large-scale streams provide another important practical conclusion: 15 edge devices and three GPU servers were sufficient to cover 20 CCTV streams. In this arrangement, an edge node based on Jetson AGX Orin covered two cameras in real time, while the server was activated only when an event was present. This result shows that shifting the central server from constant high load to event-based activation is appropriate for security systems. [43]

Table 3

Generalized view of experimental results reported in the literature

Source	Test conditions	Architecture	Main quantitative result
EVA, 2023	Real traffic streams	Edge–server	97.10% accuracy, 0.08 normalized traffic
EVA, 2023	Real traffic streams	Server-only	98.60% accuracy, 0.97 normalized traffic
Song and Nang, 2024	20 streams, “assault”	Edge-only	294 s, F1=0.848
Song and Nang, 2024	20 streams, “assault”	Server-only	1040 s, F1=0.889
Song and Nang, 2024	20 streams, “assault”	Edge–server	314 s, F1=0.904
Song and Nang, 2024	20 streams, “fall”	Edge–server	314 s, F1=0.942
Ali et al., 2024	Raspberry Pi 4 B	Edge-only	79.90% accuracy, 32.21 frames per second
Abu Awwad et al., 2024	Low-light scene	Edge-only	>20% accuracy improvement, ~1 s response
Ugli et al., 2023	Hierarchical edge	Edge–server	32.1% less GPU memory; ~3 s reload delay prevented

The data in Table 3 reveal two important tendencies: first, a purely centralized mode often loses in terms of network and time; second, a purely local mode may be limited in terms of accuracy for certain complex classes. The most favorable result corresponds to a multi-stage hybrid architecture, in which speed is close to edge-only while accuracy is higher than, or at least equal to, server-only. [44]

We now calculate integral efficiency on the basis of the mathematical model presented above. For the meta-analytical scenario, the following normalized values were adopted: for the centralized architecture, $Q_d = 0{,}91175$ (the average server-only F1 obtained from Song and Nang's day-and-night indicators), $\hat{T}_q = 1$ (1040 seconds as the baseline value), $\hat{L}_t = 0{,}97$ (the EVA server-only mode), and $\hat{G}_s = 1$ (baseline load). For the proposed architecture, the values are $Q_d = 0{,}926$ (integrated day-and-night average F1), $\hat{T}_q = 314/1040 \approx 0{,}302$, $\hat{L}_t = 0{,}08$ (the EVA edge-assisted mode), and $\hat{G}_s = 0{,}679$ (based on 32.1% memory saving). Although these values were not obtained from one and the same task, they provide a sufficient meta-analytical approximation for evaluating the overall efficiency of edge–central distribution. [45]

$$E_{markaz} = 0{,}35 \cdot 0{,}91175 + 0{,}25(1-1) + 0{,}25(1-0{,}97) + 0{,}15(1-1) = 0{,}3266$$

$$E_{taklif} = 0{,}35 \cdot 0{,}926 + 0{,}25(1-0{,}302) + 0{,}25(1-0{,}08) + 0{,}15(1-0{,}679) = 0{,}7768$$

Thus, according to the integral efficiency indicator, the proposed multi-stage architecture is approximately 2.38 times superior to the centralized baseline architecture. This result is not a separate laboratory measurement; rather, it is an estimate obtained by integrating several reliable experimental sources into a single mathematical indicator. Nevertheless, it clearly confirms the scientific conclusion at the architectural level: the strategy of evaluating the threat locally first and transmitting only filtered information to the center significantly increases overall system efficiency.

If 'processing time for 20 streams' is depicted as a bar chart, the centralized server-only bar will be the highest at 1040 seconds, the edge-only bar will remain at the lowest point at 294 seconds, and the proposed architecture will be very close to edge-only at 314 seconds. However, if the F1 indicator is plotted on the second axis, the integrated bar will be above the other variants with F1=0.904 and 0.942. The meaning of this graph is that the proposed model increases accuracy while almost preserving speed. [46]

In a line graph of normalized network-load values, the server-only mode is located at 0.97, whereas the EVA edge-assisted mode is located at 0.08. The large distance between these two points shows that the selective transmission mechanism sharply reduces traffic. If an accuracy line is added to the same graph, it becomes clear that the difference between 98.60% and 97.10% is very small, whereas the gain in traffic is very large. [39]

In a graph of day-and-night F1 indicators, the curve of the integrated architecture remains relatively stable in

the range of 0.902–0.968, while the edge-only and server-only curves decline more sharply in some cases. Especially in nighttime assault and daytime assault scenarios, the curve of the integrated solution is higher than both separate methods. This graph indicates that the proposed model is more adaptable to changes in external conditions. [47]

Discussion

The obtained results show that real efficiency in a video surveillance system is determined not by 'maximizing accuracy' alone but by a useful balance among accuracy, speed, and transmission cost. The centralized server-only approach often offers high accuracy, but this is almost always achieved at the expense of network load and response time. The edge-only mode provides minimal traffic and low latency; however, in some complex classes or difficult conditions, precision is insufficient. The proposed multi-stage model occupies the optimal point between these two extremes. Both the generalized results in the literature and the integral efficiency value of 0.7768 confirm this conclusion. [48]

From the standpoint of discussion, the most important principle is task layering. A camera or local device should perform fast, lightweight, and high-recall stages rather than high-accuracy but expensive semantic inference. Conversely, the central server should be freed from continuous stream load and should perform high-accuracy analysis only on suspicious and dangerous fragments. In Song and Nang's experiment, the edge stage functioned precisely as a source of high recall, while the server stage functioned as a source of high precision; this practically supports the theoretical argument. [49]

Another important issue is that edge architecture should not be understood merely as 'reducing the model size'. If the local stage is limited only to detection, results may be weak in complex environments. As shown by Abu Awwad et al., preparatory stages such as image enhancement and illumination-type classification are also integral components of overall efficiency. Likewise, the resource management demonstrated by Ugli et al. is important: in some cases, when to keep a model in memory and when to release it affects efficiency more than the model itself. Therefore, edge artificial intelligence should be interpreted not as 'one model plus one device' but as a 'multi-component, resource-aware system'. [50]

The relevance of the proposed model to the field of technical security tools lies in the fact that it significantly reduces the load on the central control point by processing information at the facility itself. This approach is particularly useful in warehouses, production areas, logistics centers, educational institutions, transport platforms, and edge facilities with limited network capacity. For example, transmitting only fragments in which unauthorized entry into an area, a suspicious item, a fall, an assault, or prolonged immobility is recorded, instead of transmitting the complete raw stream, is a justified solution both theoretically and in terms of the results reported in the literature. [51]

At the same time, this approach also has limitations. First, reducing results

obtained from different articles to a single integral criterion is a meta-analytical approach and cannot fully replace simultaneous measurement of all indicators at one real test site. Second, illumination, weather, camera height, object size, and local risk scenarios vary from one location to another; therefore, risk weights and threshold values require local calibration. The third issue concerns privacy and information security: although the raw video stream is transmitted to the center less frequently, metadata and signal packets must also be managed through cryptographic protection, access control, and storage policies. [52]

Several directions appear important for future research. The first is federated approaches in which edge devices do not exchange data but synchronize model parameters. The second is to make multi-camera re-identification more stable in a local–central combination. The third is to include complex environmental conditions, such as low illumination, smoke, rain, and snow, more explicitly in the risk-assessment formula. The fourth is to incorporate energy consumption and resilience criteria into the normalized integral efficiency indicator. [34]

Conclusion

As a result of the conducted research and meta-analytical generalization, the following scientific conclusions were obtained. First, an architecture based on transmitting video surveillance streams completely to the center is not optimal for real-time security systems, because it imposes excessive requirements on network load, latency, and server resources. Second, although purely local analysis is convenient in terms of network and time, it may be limited in accuracy under complex risk scenarios. Third, it is precisely the multi-stage edge–server architecture that resolves this contradiction: the edge stage preprocesses video data, detects objects, and assesses risk level, while the central server performs deeper verification only of selectively transmitted episodes. [53]

The most important scientific result is that the proposed model combines three main functions performed on the local device—initial preparation, object detection, and risk-level assessment—into a single architecture. This novelty transforms the central server from a continuous inference center into a center for event verification and system-level management. Based on reliable experiments reported in the literature, the hybrid architecture reduced the processing time for 20 streams from 1040 seconds to 314 seconds, increased the F1 indicator for assault detection to 0.904 and for fall detection to 0.942, and produced a day-and-night average F1 of 0.926. [54]

From the standpoint of network load, the edge-assisted approach based on selective transmission reduced normalized traffic from 0.97 in the centralized mode to 0.08, thereby reducing traffic by approximately 91.8%. For low-light environments, the local preparation stage improved detection quality by more than 20%; on a resource-constrained local device, quantization provided a speed of 32.21 frames per second and 2.77 times higher energy efficiency. The hierarchical approach to server resource management reduced GPU memory consumption by up to

32.1%. These results show that the proposed model is not only theoretical but also practically grounded. [55]

According to the developed integral efficiency criterion, the centralized baseline architecture showed $E = 0{,}3266$, whereas the proposed multi-stage edge-active architecture showed $E = 0{,}7768$. Thus, from the standpoint of normalized meta-analytical assessment, the proposed model is approximately 2,38 times superior to the centralized approach in terms of overall efficiency. Therefore, the scientific novelty advanced in the article—the multi-stage intelligent detection model based on performing video preprocessing, object detection, and risk-level assessment directly on a surveillance camera or local computing device—is recommended as a scientifically grounded and practically promising direction for improving threat detection efficiency in intelligent security systems.

References;

1. Sh.Q. Shoyqulov.On the study of optical communication systems using simulators.Eurasian journal of mathematical theory and computer sciences, T. 5, Выпуск 11. Nov. 2025. P. 20–28. https://doi.org/10.5281/zenodo.17640489
2. Sh.Q. Shoyqulov.AI-enhanced Web scraping for data-driven analysis .Central Asian Journal of Multidisciplinary Research and Management Studies (CAJMRMS), Vol 2, Issue 11. Nov. 2025. P. 20-27. ISSN:3030-3540. www.in-academy.uz. https://doi.org/10.5281/zenodo.17529443
3. Sh.Q. Shoyqulov.Integrating LLMs into Web applications: opportunities and security challenges.Shoyqulov, S. (2025). Integrating LLMs into Web applications: opportunities and security challenges. Eurasian journal of mathematical theory and computer sciences (T. 5, Выпуск 6, cc. 54–60). https://doi.org/10.5281/zenodo.15755908
4. Sh.Q. Shoyqulov .AI-driven UX optimization for Web applications.Shoyqulov, S. (2025). AI-driven UX optimization for Web applications. Eurasian journal of mathematical theory and computer sciences (T. 5, Выпуск 6, cc. 46–53). https://doi.org/10.5281/zenodo.15755881
5. Sh.Q. Shoyqulov.Analysis and optimization of graphics programming in C# using Unity.«Science and innovation» xalqaro ilmiy jurnali, Volume 3 Issue 10, p.69-75. https://doi.org/10.5281/zenodo.14000841
6. Sh.Q. Shoyqulov.Main Internet threats and ways to protect against them.Евразийский журнал академических исследований, 4(10),p. 140–146. DOI: https://doi.org/10.5281/zenodo.13991390
7. Sh.Q. Shoyqulov.Using Python programming in computer graphics.«Science and innovation» xalqaro ilmiy jurnali, Volume 3 Issue 10, p.18-24, https://doi.org/10.5281/zenodo.13926022
8. Sh.Q. Shoyqulov and oth..Propagation of Non-Stationary Waves Of Transverse Displacement from a Spherical Cavity in an Elastic Half-Space.International Journal of Advanced Research in Science, Engineering and Technology, Vol. 7, Issue 4, April 2020, p.13291-13299, ISSN: 2350-0328, http://www.ijarset.com/upload/2020/april/13-shshovqulov-02-1.pdf

9. Sh.Q. Shoyqulov and oth..Methods for plotting function graphs in computers using modern software and programming languages.Published in ACADEMICIA An International Multidisciplinary Research Journal ISSN: 2249-7137, Vol. 11 Issue 6, JUNE 2021. P. 321-329, India, Impact Factor: SJIF 2021 = 7.492, https://saarj.com/academicia-view-journal-current-issue/
10. Sh.Q. Shoyqulov and oth..Computer graphics in technical disciplines .EURASIAN JOURNAL OF ACADEMIC RESEARCH (T. 4, Выпуск 10, cc. 21–27). Zenodo. https://doi.org/10.5281/zenodo.13898180
11. 11 .Sh.Q. Shoyqulov and oth..Computer graphics in the natural sciences .EURASIAN JOURNAL OF ACADEMIC RESEARCH (T. 4, Выпуск 10, cc. 12–20). Zenodo. https://doi.org/10.5281/zenodo.13898146
12. Sh.Q. Shoyqulov.The Role and Possibilities of Multimedia Technologies in Education.International Journal of Discoveries and Innovations in Applied Sciences (IJDIAS), www.openaccessjournals.eu, Volume: 2 Issue: 3 in March-2022, http://openaccessjournals.eu/index.php/ijdias/article/view/1148
13. Sh.Q. Shoyqulov.Technical and Software Capabilities of a Computer for Working with Multimedia Resources.International Journal of Discoveries and Innovations in Applied Sciences (IJDIAS), Volume: 2 Issue: 3 in March-2022, http://openaccessjournals.eu/index.php/ijdias/article/view/1147
14. Sh.Q. Shoyqulov.The text is of the main components of multimedia technologies.Academicia Globe: Inderscience Research, 3(04), 573–580. https://agir.academiascience.org/index.php/agir/article/view/684, ISSN: 2776-1010, SJIF: 5.653, Impact Factor: 7.425, https://doi.org/10.17605/OSF.IO/VBY8Z
15. Sh.Q. Shoyqulov and oth..PHP is one of the main tools for creating a Web page in computer science lessons.Texas Journal of Engineering and Technology, Vol. 9 (2022): TJET, p.83-87, ISSN (Online): 2770-4491, SJIF Impact Factor (2022): 5.577, https://zienjournals.com/index.php/tjet/article/view/2000/1689
16. Sh.Q. Shoyqulov and oth..Multimedia surveillance cameras and their features in using.International Journal of Innovations in Engineering Research and Technology (IJIERT), Vol 9, No 10 (2022), p.29–34. https://repo.ijiert.org/index.php/ijiert/article/view/3379, doi.org/10.17605/OSF.IO/4EV75
17. Canel, C., Kim, T., Zhou, G., Li, C., Lim, H., Andersen, D. G., Kaminsky, M., Dulloor, S. R. Scaling Video Analytics on Constrained Edge Nodes. Proceedings of the 2nd SysML Conference, 2019. [72]
18. Lyu, Z., Li, J., Li, B., Zhang, Y., va boshq. A Surveillance Video Real-Time Object Detection System Based on Edge-Cloud Computing. Applied Sciences, 2022, 12(19), 10128. [73]
19. Wang, X., Tang, Z., Guo, J., Meng, T., Wang, C., Wang, T., Jia, W. A Comprehensive Survey on On-Device AI Models. ACM Computing Surveys, 2025. [74]
20. Glenn, J. Public Safety Communications Research Division Impact Report. NIST Special Publication 1248, 2020. [75]