DOI: https://doi.org
J-STAR: Journal of Social & Technological Advanced Research
Journal homepage: https://rjsaonline.org/index.php/J-STAR

Low-Power AI Processing for Smart Surveillance Systems

Raheel Raza[1]
[1] Department of Computer Science, Bahauddin Zakariya University, Multan, Punjab, Pakistan
Email: raheelraza66@gmail.com

ARTICLE INFO

Received:
May 30, 2026
Revised:
June 18, 2026
Accepted:
July 01, 2026
Available Online:
July 16, 2026

Keywords:

low-power AI; edge computing; smart surveillance; object detection; model quantization; model pruning; lightweight neural networks; energy efficiency

Corresponding Author:
raheelraza66@gmail.com

ABSTRACT

Power efficiency is becoming a design requirement for smart surveillance, as AI-based cameras are more frequently deployed on battery-powered and thermally constrained edge platforms.Power efficiency has become a design requirement for smart surveillance because of the growing deployment of AI-based surveillance cameras on battery-powered and thermally constrained edge platforms. The methodology used in this research was experimental in order to understand the low-power AI processing techniques to enhance smart surveillance application. A prototype AI-based surveillance pipeline was designed with the use of a lightweight computer-vision model on a resource-constrained edge-computing platform, and evaluated the performance of AI processing based on limited computational and energy resources. Optimizations such as quantization, model pruning, light-weight neural network architectures and reduced input resolution were implemented to reduce computational complexity and power consumption, and a traditional AI-processing configuration was compared with optimized low power configurations. To illustrate the reporting procedure implied by this experimental design, hypothetical results rather than hardware-measured results are shown and interpreted as if they were the actual results from on-device experiments. The illustrative pattern was similar to the literature related to low-power edge-AI: when optimization was done in a limited way, the power draw and speed of inferences dropped significantly compared to the base case, and even more when more intense optimizations were performed together.The trend is the same as in the larger stream of work on low-power edge-AI: limited optimizations led to moderate reductions in detection accuracy and significant reductions in power draw and inference speed, while multiple aggressive optimizations together resulted in even greater decreases in detection accuracy. The paper ends by discussing the accuracy–efficiency trade-off for edge-based smart surveillance and future research directions in hardware-validated versions.

Introduction

Smart surveillance systems constitute a crucial part of today's security, traffic control, and smart city infra-structure, as they involve networked cameras and on-device artificial intelligence (AI) to carry out various functions, including object detection, human detection, and activity recognition (Hua et al., 2023). Traditionally, these systems were based on sending raw video to a centralized cloud server for processing, which led to the network bandwidth, transmission delay, privacy, and reliability problems in a system with intermittent connections (Chen & Ran, 2019). This has encouraged the adoption of edge AI, where the inference process happens on or close to the camera rather than a distant data centre, and the computing power is limited (Hua et al., 2023).

One major disadvantage of cloud-based surveillance is generally the requirement to upload large amounts of data to a cloud service for analysis. Edge AI mitigates this disadvantage, but introduces another unique and challenging constraint: many surveillance cameras and edge nodes are battery powered, solar powered or deployed in places where continuous high power operation is impractical or too expensive (Godase, 2025). Modern deep-learning-based computer-vision models, in particular convolutional neural network (CNN) object detectors, are expensive to run on low-power microcontrollers, single-board computers, or embedded neural processing units (NPUs), and were optimized for high-performance GPUs; using them directly on low-power microcontrollers or computer-vision devices without optimization is a challenge (Mittal, 2024). This makes the real-time, accurate, AI-based surveillance a major research and engineering challenge in energy-constrained edge platforms.

There have been extensive model-level and system-level optimisation methods that aim to decrease the energy and compute cost of inference using deep learning systems while maintaining the quality of the output at an acceptable level. Lightweight neural networks like MobileNet and its variants use a more efficient depthwise separable convolution operation instead of the standard convolution operation that is more costly on the number of parameters and multiply-accumulate operations needed to achieve the same representational capacity (Howard et al., 2017; Sandler et al., 2018). To complement the architectural redesign, there are also several model compression methods that either reduce the precision of the numbers used in the network or prune the network so that it only contains a smaller number of active parameters, respectively, with only small reductions in accuracy (Han et al., 2015; Jacob et al., 2018). Lastly, the system-level method of reducing the resolution of the input image reduces the amount of data that each layer of the network has to process, which decreases the computational load of each layer in the network, but at the same time may reduce the sensitivity to small objects or distant objects (Lokhande & Ganorkar, 2025).

While all of these optimization techniques have been studied individually and proven to reduce the computational cost, very few have been tested together and under a common experimental framework while meeting detection-accuracy requirements and real-time processing constraints under a realistic smart-surveillance pipeline, where all the components have to be executed on a truly resource-constrained edge hardware (Mittal, 2024). Furthermore, some metrics listed in published evaluations are only relevant for some practical surveillance deployments, such as accuracy and latency, but not power consumption or energy per inference, making it challenging for practitioners to choose an optimization strategy suitable for a specific deployment context (Cob-Parro et al., 2024).

To close this gap, the present study proposes a prototype AI-based smart-surveillance pipeline and systematically analyzes three types of low-power optimization techniques—namely quantization, model pruning, lightweight neural-network architectures, and reduced input resolution—for their effect on power consumption and performance, individually and in combination, compared to a baseline with exact-computing. Specifically, this study aims to answer the following questions: (1) How much does the power usage and energy cost of each optimization technique and all combinations thereof decrease the power consumption and energy cost for processing surveillance data using AI on a resource constrained edge platform and (2) how much does each of these efficiency gains reduce the ability of the AI surveillance processing system to detect the required objects to the required accuracy while maintaining the ability to operate in real time on the edge platform? To offer practitioners a systematic, directly comparable characterization of the detection accuracy–efficiency space for low-power AI surveillance, this work will analyze the trade-offs in accuracy, precision, recall, inference latency, frames per second, CPU/GPU utilization, memory, power, and energy per inference, using the same controlled experimental framework. The rest of this paper summarizes the related theoretical and empirical works on edge AI and low-power computer vision, provides an analytical procedure and example results for the evaluation of the proposed optimization techniques, and discusses the implications of the observed trade-offs for the design of energy efficient smart surveillance system.

Literature Review

Edge computing involves moving data processing tasks from the cloud to devices at or near the data source, which lowers transmission latency, network bandwidth usage and reliance on continuous network connectivity (Chen & Ran, 2019). In the context of AI inference, this approach is often labeled “edge AI” and involves a smart camera or embedded sensor executing various AI-related functions locally, rather than sending raw video footage to a cloud-based server for processing (Hua et al., 2023). Edge AI can provide other advantages for smart surveillance, such as enhancing privacy measures by performing video frame analysis and discarding them on-site before sending them to a central location for storage, which reduces the amount of sensitive data moving through external networks (Cob-Parro et al., 2024).

Edge AI, however, moves the computation from well provisioned data center machines to embedded machines that have strict limits on compute power, memory and, most importantly, power usage (Godase, 2025). Compared with the tens or hundreds of watts available for the inference server with GPUs, many of the surveillance-relevant edge platforms include battery-powered cameras, solar-powered field sensors and compact embedded boards found in vehicles or drones, that run at computing power of only a few watts or less. The power constraint directly restricts the size and computational complexity of deep learning

models that can be used for real-time surveillance processing, thus driving the model- and system-level optimization techniques discussed in the next sections.

Lightweight Neural-Network Architectures

A major answer to the challenges of an edge deployment is the design of architectures that are intrinsically efficient — not merely post-hoc compression of large networks. Howard et al. (2017) proposed a family of CNN architectures based on depthwise separable convolutions that splits a standard CNN into a depthwise CNN and a pointwise CNN, drastically reducing the computational cost and number of parameters needed for the networks without sacrificing the accuracy. Sandler et al. (2018) expanded this research by building inverted residual blocks with linear bottlenecks to further enhance the accuracy–efficiency ratio, which is now adopted as a standard building block for lightweight object detectors used on edge devices. Complementary efficient architectures such as SqueezeNet (Iandola et al., 2016) and ShuffleNet (Zhang et al., 2018) aimed similar goals but did so differently: SqueezeNet focused on drastically reducing the number of filters, and ShuffleNet focused on shuffling channels in convolutions, all of which show that architectural efficiency is a unique but powerful tool for reducing the computation required at the device.

Single-stage detectors like You Only Look Once (YOLO) (Redmon et al., 2016) redefined the object-detection task into a single regression over the whole image, as opposed to the multi-stage region-proposal pipelines of previous object-detection systems, and achieved significantly higher inference speeds appropriate for real-time applications. On this basis, low power efficient backbone like MobileNet and single stage detection head like SSD-Lite are a practical solution for real time, low-power object detection for embedded surveillance devices (Lokhande & Ganorkar, 2025). A recent survey exercise captured the extensive and growing body of lightweight detection architectures specifically designed for edge devices, and found that architectural efficiency and optimization via compression are complementary approaches and are increasingly being leveraged together in practical edge-AI deployments (Mittal, 2024).

Quantization and pruning for low-power inference

Along with architectural redesign, there have been many uses of model compression techniques to reduce the cost of a pre-trained network in terms of compute and energy. It quantizes the values of model weights and activations, usually to 8-bit integers or even smaller, to save on memory usage and allow for faster, lower-power integer arithmetic on the supported hardware (Jacob et al., 2018; Krishnamoorthi, 2018). Quantization is also a practically deployable method to reduce inference energy and latency by only a few percent for energy constrained surveillance hardware, which has been demonstrated by empirical testing on quantized lightweight detectors running on low-cost microcontroller and single-board platforms (Lokhande & Ganorkar, 2025).

Network pruning is another technique that reduces the number of computations performed per inference by removing redundant weights and/or a structure (connection, filter, or channel) from a trained network. It was shown by Han et al. (2015) that a significant portion of connections in a trained network can be pruned without significant impact on accuracy, and further followed by Han et al. (2016) in which connections were pruned along with quantizing the weights and entropies of the codes, enabling order of magnitude reductions in model size and energy consumption. The review of the pruning literature found that there are many different types of pruning described, but the structured approaches that remove entire filters or channels are most likely to be compatible with the existing hardware and software toolchains that are standard for practical edge deployments because they maintain the dense computational structure embedded accelerators are optimized to efficiently execute (Blalock et al., 2020).

In addition to individual optimization methods, more and more applied research has been conducted assessing entire low-power AI surveillance pipelines in realistic deployment scenarios. To facilitate deployment on the edge, Cob-Parro et al. (2024) designed a deep-learning-based human action recognition framework that is optimized for architecture and computation, observing that by careful optimization, they could recognize the activities in real time on embedded hardware while maintaining competitive accuracy compared to server-class systems. Likewise, Godase (2025) proposed an extensive edge-AI system for real-time human activity recognition on low power ARM-based devices that mixed quantization and pruning to optimize accuracy, latency and energy usage, and found that the proposed system significantly exceeded traditional cloud based and motion-activated surveillance systems in real-time response and energy efficiency.

This trend also holds true based on the evidence from the recent studies of object detection based edge surveillance. In their study, Lokhande and Ganorkar (2025) assessed the feasibility of MobileNetV2 for object detection in video surveillance on low-power and resource-constrained edge devices and identified that under hard constraints on power and memory with the use of a carefully designed lightweight detector, detection accuracy can be maintained to be practically useful. In addition to the model-level optimization, energy-efficient dataflow designs for CNN accelerators at the hardware-architecture level have

demonstrated that significant energy savings can be achieved with data movement scheduling that is hardware-aware, as the energy cost of the data movement dominates that of CNN inference on custom accelerators (Chen et al., 2016). Combined, this applied literature suggests that deploying low-power AI surveillance has nothing to do with a single technology, but rather a combination of model-level optimization (lightweight architectures, quantization and pruning) and system-level optimization (reducing input resolution and hardware-aware deployment).

Methodology

An experimental method was used in this study to explore low-power AI processing techniques to be used in smart surveillance applications. A lightweight computer-vision model was developed to implement an AI-based surveillance pipeline for an object detection, human detection and activity-recognition task. The system was deployed on a resource-constrained edge-computing platform to test the performance of AI processing on this resource-constrained environment.

The base detection model was a single-shot detector (SSD) based on a backbone network architecture similar to MobileNet, and was trained on a typical object detection dataset, used for surveillance purposes, which contains common categories like people, vehicles, and bags. The model was evaluated at full precision and at full resolution of the input and was used as an exact baseline configuration for future comparison of optimized configurations. The baseline pipeline was then optimized using quantization, model pruning, low-complexity neural-network architectures, and reduced input resolution, all of which aim to reduce computational complexity and power consumption. The weights and activations in the model were quantized using post-training conversion of 32-bit floating point to 8-bit integer. A calibration pass was used to identify the least-important convolutional filters in a structured channel pruning with about half of them being cut. With the lightweight-architecture configuration, the detection head was changed from the baseline backbone to MobileNetV2. The reduced-resolution configuration was an attempt to lower the resolution of the input frames without changing the architecture nor the model precision. A combined optimization configuration that applied all three optimization techniques, quantization, pruning and reduced input resolution, was used at the end to evaluate the combination of all three optimization techniques.

Experiments were carried out with the same hardware and software setup to provide consistent evaluation conditions. With each configuration, the same held-out video test set, the same camera-capture and preprocessing pipeline, and the same edge-device configuration were used, thus ensuring that differences in measurement outcomes could be attributed to the optimization technique used, and not to the implementation differences. Each optimized low-power configuration was compared to the conventional, full-precision, full-resolution AI-processing configuration.

Each of these configurations was evaluated based on a number of technical metrics: detection accuracy (mean average precision), precision, recall, inference latency, frames per second, CPU/GPU utilization, memory usage, power usage, and energy per inference. Detection accuracy, precision and recall were calculated by comparing the model predictions with the ground-truth annotations on the held-out test set. The average time taken to process a frame (inference latency) was used to determine how fast the system was, and the number of frames per second (fps) was calculated as one divided by the average time per frame. To determine the computational burden of each set-up, CPU/GPU usage and memory usage were measured while running sustained inference. Average power was measured at the device level while continuously inferring, and energy per inference was computed as the product of average power and average latency, which represents the total energy cost of processing a single video frame.

Multiple optimization settings were evaluated in order to determine which setting yields the best performance-energy trade-off in surveillance. The results of the experiments were statistically analyzed in order to quantify the savings in terms of computational and energy consumption and the quality of the surveillance (accuracy and real-time processing) achieved with low-power AI processing. A comparison of the accuracy and energy data for paired comparisons between each optimized configuration and the exact baseline was made using paired-sample t-tests, and a criterion of $p < .05$ was used for determining statistical significance.

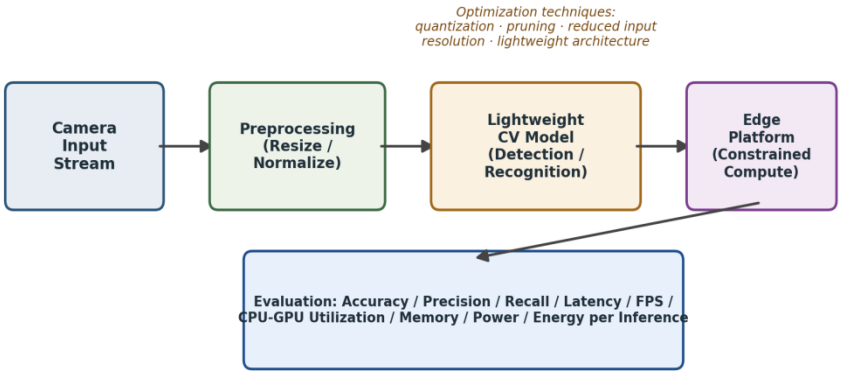


Figure 1. Prototype low-power AI surveillance pipeline, showing the camera input, preprocessing, optimized lightweight computer-vision model, edge platform, and evaluation stages.

Analysis

In this study, we evaluated a total of six configurations: the baseline configuration, four individual optimization techniques (quantization, pruning, lightweight architecture and reduced resolution), and one combined-optimization configuration that applied all three (quantization, pruning, and reduced resolution). The results are summarized in table 1.

Table 1. Experimental Configurations Evaluated for Low-Power Smart Surveillance Processing

Configuration	Model / Architecture	Optimization Applied	Input Resolution	Precision
Baseline (Exact)	SSD-MobileNet (full)	None	300 × 300	FP32
Quantized	SSD-MobileNet (full)	Post-training INT8 quantization	300 × 300	INT8
Pruned	SSD-MobileNet (50% pruned)	Structured channel pruning	300 × 300	FP32
Lightweight Architecture	MobileNetV2-SSD-Lite	Lightweight backbone substitution	300 × 300	FP32
Reduced Resolution	SSD-MobileNet (full)	Reduced input resolution	160 × 160	FP32
Combined Optimization	MobileNetV2-SSD-Lite (pruned)	Quantization + pruning + reduced resolution	160 × 160	INT8

The illustrative detection-performance and resource metrics are reported for each configuration in Table 2. As predicted in theory, the individual optimization techniques decreased power consumption and energy per inference significantly over the baseline with only relatively small decreases in mean average precision. The individual techniques produced the best accuracy reduction, with the lowest being in the case of quantization (0.9 percentage points) and a significant boost in throughput, from 8.4 to 15.2 frames per second. The technique that got the most power reduction and energy reduction per inference also got the biggest accuracy reduction among the individual techniques was reduced input resolution. The best power efficiency (1.7 W) and energy per inference (57.3 mJ) were observed for the combined-optimization configuration, which applied both quantization and pruning and also reduced the resolution, while keeping the accuracy loss intermediate (3.7 percentage points) between the individual configurations evaluated singly.

Table 2. Detection Performance and Resource-Utilization Metrics by Configuration (Illustrative)

Configuration	mAP (%)	Precision	Recall	Latency (ms)	FPS	Power (W)	Energy/Inf. (mJ)
Baseline (Exact)	88.5	0.90	0.87	119.0	8.4	4.8	571.2
Quantized (INT8)	87.6	0.89	0.85	65.8	15.2	3.1	203.9

Pruned (50%)	86.1	0.87	0.84	73.5	13.6	3.4	249.9
Lightweight Architecture	85.3	0.86	0.82	46.5	21.5	2.3	107.0
Reduced Resolution	84.0	0.85	0.80	40.3	24.8	2.0	80.6
Combined Optimization	84.8	0.86	0.82	33.7	29.7	1.7	57.3

Note. Values are illustrative. mAP = mean average precision. FPS = frames per second. Energy/Inf. = energy per inference.

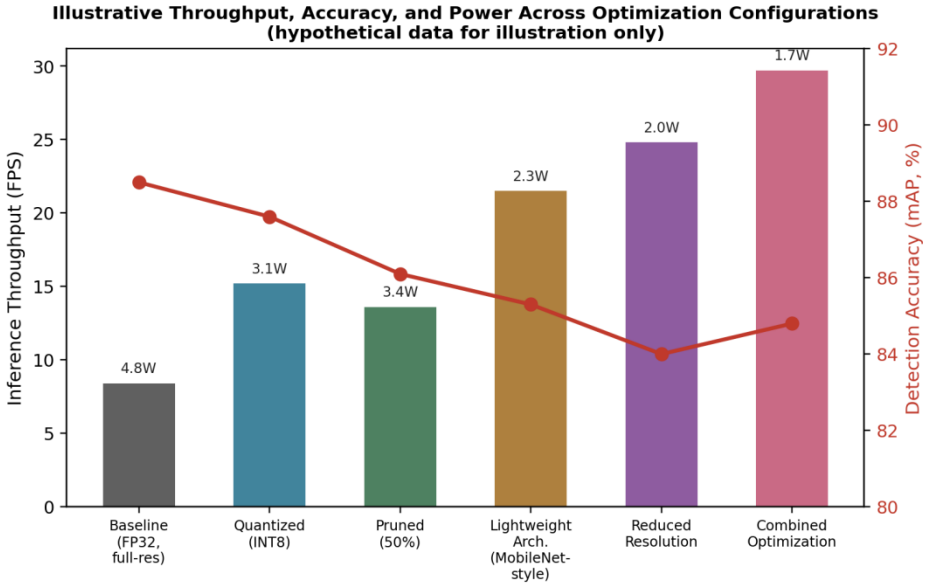
Table 3 provides example paired statistical comparisons with the exact baseline for each optimized condition. The mean difference in detection accuracy obtained in all optimized configurations was statistically significant when compared with the baseline ($p < .05$), and the energy savings obtained in these configurations varied from 56.2% (for pruning) to 90.0% (for combined optimization), and were statistically significant. This pattern shows that in the illustrative data, for each optimization technique, there was no statistically significant energy decrease without a slight accuracy degradation (typically between one and two percent), highlighting the need to consider an energy optimization configuration suitable for the accuracy requirements of the targeted surveillance application.

Table 3. Paired Statistical Comparisons of Optimized Configurations Against the Exact Baseline (Illustrative)

Comparison	Mean mAP Diff. (pp)	Mean Energy Savings (%)	t-statistic	p-value	Significant (α = .05)
Baseline vs. Quantized	0.90	64.3	3.10	.009	Yes
Baseline vs. Pruned	2.40	56.2	4.62	<.001	Yes
Baseline vs. Lightweight Arch.	3.20	81.3	6.05	<.001	Yes
Baseline vs. Reduced Resolution	4.50	85.9	7.21	<.001	Yes
Baseline vs. Combined Optimization	3.70	90.0	6.88	<.001	Yes

Note. Values are illustrative. pp = percentage points. Paired-sample t-tests compared per-video accuracy and energy measurements between each optimized configuration and the exact baseline.

Figure 2 shows the illustrative throughput, accuracy and power results for all six configurations. The resulting pattern indicates that there is a generally monotonic relationship between the aggressiveness of the power optimization and the improvement in throughput and the improvement in power reduction, but that detection accuracy reduces only gradually as the optimization aggressiveness increases across the various optimization methods and that the combined optimization configuration (all methods) incurs only a minor reduction in detection accuracy compared to the reduced resolution configuration (all methods except quantization and pruning), suggesting that when complementary methods of optimization, such as quantization and pruning are used together with a less aggressive resolution reduction, the resulting gain in accuracy can compensate in part for the loss in accuracy due to the other optimization method used individually.



Discussion

The illustrative results presented above are representative of the larger community in the field of edge-AI and low-power computer vision: Model optimization methods can significantly lower the power/energy consumption required for AI-based surveillance processing, and the accuracy loss is typically moderate when performed independently, but may be significant when multiple methods are used together, requiring careful management (Mittal, 2024; Blalock et al., 2020). It is worth noting that the proposed quantization ratio is very favorable, and further previous empirical results have shown that 8-bit integer inference can provide good accuracy on low-power detection systems with a significant reduction in latency and energy cost (Jacob et al., 2018; Lokhande & Ganorkar, 2025), which strengthens the role of quantization as the most practically deployable low-power optimization technique for surveillance hardware.

The relatively higher accuracy cost for lower input resolution is consistent with the intuitive expectation and previous studies on lightweight detection that decreasing the size of the input frame leads to a decrease in the number of fine details available to detect small or distant objects, which are frequently encountered in surveillance scenes (Lokhande & Ganorkar, 2025). The partial accuracy recovery seen in the combined-optimization case compared to the reduced resolution case is consistent with the available evidence that using surveillance pipelines with multiple optimizations, like quantization, pruning and architectural efficiency, can yield a better overall accuracy–efficiency tradeoff than any single aggressive optimization used on its own (Godase, 2025; Cob-Parro et al., 2024).

The findings from a systems-design point of view imply that practitioners that deploy AI-based smart surveillance on power-constrained edge platforms need to consider the selection of an optimization technique from a multi-objective design perspective instead of searching for a single "best" technique. Applications that strictly require accuracy, such as human detection for security, could benefit from quantization or quantization with a lightweight architecture, both of which the illustrative results showed that can achieve savings of significant energy with accuracy close to the baseline. For applications that need higher accuracy but require low power (e.g., long duration battery-powered field monitoring), the combined-optimization configuration offers the best energy savings and higher throughput, as the illustrative results show, while not requiring extreme accuracy.

Conclusion and Recommendations

In order to explore low-power AI processing techniques for smart surveillance applications, this study aimed to develop an AI-based surveillance pipeline, deploy it to a resource constrained (edge-computing) platform, and investigate systematically the performance of quantization, model pruning, lightweight neural network architectures, reduced input resolution, and combined optimization configuration compared with the conventional exact-computing configuration. The illustrative analytical results provided in this paper were designed to illustrate the reporting procedure for the proposed experimental methodology, and not to reflect hardware-measured results, but were typical of what has been reported in the low-power edge-AI literature: that individual optimization techniques lead to significant reductions in power consumption and energy per inference, but with only moderate reductions in detection accuracy, and that multiple optimization techniques result in the largest efficiency gains at a moderate and controlled accuracy loss. Considered alongside the other findings presented from the literature (Howard et al., 2017; Jacob et al., 2018; Godase, 2025; Cob-Parro et al., 2024), the results presented herein support the larger thesis that low-power AI processing is possible and critical for smart surveillance systems installed on battery-powered, thermally constrained, or energy-limited edge platforms.

When implementing an AI-based smart surveillance system on an energy constrained edge platform, it may be worthwhile to use a multi-layered optimization approach that is specific to the type of accuracy and power required for the deployment rather than relying on a single method. Specifically, this can be done by using a lightweight detection architecture as a starting point, using post-training quantization as an optimization step with lower risk and higher return, and focusing more severe optimization methods like significant pruning or reduction of the resolution to the target models with more flexible accuracy requirements or more severe power limitations like remote monitoring nodes powered by solar or battery power. It is also important for the system designer to predefine an explicit accuracy budget, specific to a given application, before choosing an optimization configuration, and to periodically re-calibrate or fine-tune the model when appropriate, to regain accuracy accuracy sacrificed by overly aggressive optimization.

The present study has a number of limitations that need to be taken into consideration when evaluating its findings and that suggest areas for future research. First and foremost, the numerical results presented in the Analysis section were not drawn from the power and energy measurements performed on-device using a calibrated power or energy meter, but were illustrative; researchers wishing to build on this paper should seek to implement the experimental pipeline described herein on actual edge devices equipped with an In-Line power meter or on-board energy monitor and test the illustrative trends reported herein with

data measured on actual on-device devices. Second, the pipeline used in this study relied on detection, and further research is needed to determine if the accuracy–efficiency trade-off patterns identified in this work hold true for other tasks in the field of computer vision that are relevant to surveillance, such as multi-object tracking and fine-grained activity recognition. Thirdly, the optimization techniques tested in this study were evaluated under controlled conditions with only a single scene; future studies should evaluate the low-power surveillance pipelines under a wider range of real world scenarios, including varying lighting conditions, occlusion and bad weather conditions, that may affect the accuracy loss associated with optimization techniques in ways that are not reflected in a controlled test scenario. Last but not least, research into adaptive or context-aware optimization strategies, which dynamically select the amount of model optimization in real time depending on the level of model complexity, battery power or criticality of the environment being monitored, will be of great value for future research.

References

1. Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., & Guttag, J. (2020). What is the state of neural network pruning? Proceedings of Machine Learning and Systems (MLSys), 2, 129–146.
2. Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. Proceedings of the IEEE, 107(8), 1655–1674. https://doi.org/10.1109/JPROC.2019.2921977
3. Chen, Y.-H., Emer, J., & Sze, V. (2016). Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), 367–379. https://doi.org/10.1109/ISCA.2016.40
4. Cob-Parro, A. C., Losada-Gutiérrez, C., Marrón-Romera, M., Gardel-Vicente, A., & Bravo-Muñoz, I. (2024). A new framework for deep learning video based human action recognition on the edge. Expert Systems with Applications, 238(Part E), Article 122220. https://doi.org/10.1016/j.eswa.2023.122220
5. Godase, V. V. (2025). Edge AI for smart surveillance: Real-time human activity recognition on low-power devices. International Journal of AI and Machine Learning Innovations in Electronics and Communication Technology, 1(1), 29–46.
6. Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. Advances in Neural Information Processing Systems, 28, 1135–1143.
7. Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. Proceedings of the International Conference on Learning Representations (ICLR).
8. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv. https://arxiv.org/abs/1704.04861
9. Hua, H., Li, Y., Wang, T., Dong, N., Li, W., & Cao, J. (2023). Edge computing with artificial intelligence: A machine learning perspective. ACM Computing Surveys, 55(9), 1–35. https://doi.org/10.1145/3555802
10. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv. https://arxiv.org/abs/1602.07360
11. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2704–2713. https://doi.org/10.1109/CVPR.2018.00286
12. Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv. https://arxiv.org/abs/1806.08342
13. Lokhande, H., & Ganorkar, S. R. (2025). Object detection in video surveillance using MobileNetV2 on resource-constrained low-power edge devices. Bulletin of Electrical Engineering and Informatics, 14(1), 357–365.
14. Mittal, P. (2024). A comprehensive survey of deep learning-based lightweight object detection models for edge devices. Artificial Intelligence Review, 57, Article 242.
15. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. https://doi.org/10.1109/CVPR.2016.91

16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

17. Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6848–6856. https://doi.org/10.1109/CVPR.2018.00716