Saudi Journal
of Humanities

المجلة السعودية للعلوم الإنسانية
Saudi Journal of Humanities

Twelfth edition
Volume (4) Issue (3)
(1-30) 2026

Energy-Aware Edge AI Framework for Real-Time Object Detection Using Traffic Cameras in Baghdad City

Mustafa Abdulkhader Jassim
(Master of Science in Computer Science)

Published on: Published online on 10, August 2026

Abstract :
The high rate of urbanization of Baghdad City has augmented the level of traffic congestion challenges that require smart real-time monitoring systems that can work within limited computational and energy budgets. The presented paper suggests an Energy-Considerate Edge AI Framework of real-time object detection based on traffic cameras mounted in the road network of Baghdad. The framework combines a lightweight YOLO-based detector with an adaptive Energy-Aware Scheduling Module (EASM) that will dynamically adapt inference frequency based on real-time power telemetry and an estimate of scene complexity. Experimental tests of traffic images at various Baghdad crossing points prove that the proposed framework yields an average Precision (mAP) of 87.3% and less energy consumption (up to 43% less than traditional cloud-offloaded systems). The system maintains an average throughput rate of 28 frames per second on embedded edge hardware, and meets with real-time standards of operation at varying levels of traffic density and lighting load. The contribution of this work is a blueprint of smart city traffic surveillance that is scalable and practically deployable, specifically in the context of the constraints of resources in urban areas of developing countries.

Keywords:

Edge AI, YOLO, Object Detection, Energy Efficiency, Traffic Monitoring, Baghdad, Smart City, Real-Time Inference

1. Introduction

The nature of the challenges that face urban traffic management in developing megacities like Baghdad are compounded by almost an unprecedented increase in population, aging road infrastructure, and a highly heterogeneous warren of vehicle types that include personal cars, minibuses, motorcycles, and heavy freight carriages that can be found on the same roads as this was not what was originally planned.
Conventional surveillance designs that pipe encoded video feeds to centralized data centers are characterized by bandwidth cost that is prohibitive, latency that is fundamentally non-compliant with real-time response needs, and single points of failure that are critically harmful to the overall system reliability in already overstretched municipal networks. Another interesting alternative to the distributed intelligent system is the introduction of edge computing as a paradigm shift in distributed intelligence that provides a type of computation that is executed near the data source, with transmission overhead reduced significantly, and allowing decision loops of less than one second, required by responsive traffic management [24].
Over the last few years, artificial intelligence in the edge has come to the level of incredibly high maturity, both due to the simultaneous further development of model compression methods, neural architecture search, and the creation of specific inference accelerators that are available at very low prices in the form of a system-on-chip, which can be found even in the budgets of the municipalities. Detection of objects, as an inherent computer vision task that the traditional intelligent traffic application is built on vehicle counting to incident-detecting, has seen an enormous boost in the YOLO family

of algorithms, which re-casts the detection task as a regression problem, and sacrifices marginal accuracy to achieve transformative improvements in inference time making it practically possible to deploy edges in a real-time application [28]. Placed on edge devices that are co-located with the traffic camera infrastructure, such models can enable a city to do vehicle classification, density intention, and anomaly discovery with the low latency costs of a round-trip through the clouds, as well as the bandwidth costs of streaming raw high-definition video [29].

The city of Baghdad is an exceptionally interesting and under investigated case study of edge-deployed traffic intelligence. Its road system comprises multilane arterial highways, heavily congested downtown crossroads, and unstructured traffic patterns that do not especially match the structured driving conditions upon which most publicly available benchmark datasets were initially simulated, giving it a domain gap which directly affects the performance of detection models when applied out-of-the-box. Moreover, there are occasional periods of supply interruption of different lengths and different frequency of the electrical grid of Baghdad in each city district, which makes energy-conscious system design not only an efficiency optimization but a technical requirement of the system functioning to have a permanent surveillance coverage [14]. A grid-aware framework that consumes less energy and minimizes the amount of power used per unit of time extends the lifespan of camera nodes in periods of grid instability and incurs significantly lower costs in terms of overall deployment of hundreds of geographically dispersed nodes.

Although there is an increasing amount of literature on edge-deployed detection system, there remains an obvious and substantial gap in models that co-design model-architecture choice, hardware deployment policy, and dynamic-energy management as a coherent and consistent system and not as independent components dimension is optimized independently. The majority of previous literature either maximizes the accuracy of detection or the speed at which the inference speed, but rarely both and all in the same end-to-end deployable pipeline in a realistic municipal infrastructure [7]. This gap is covered in this paper in four main contributions:

I. A combined inference pipeline based on a lightweight YOLO variant and a new EASM that dynamically adapts inference load balance to available power constraint and the observed level of scene activity.
II. A Baghdad-specific traffic data that was recorded at the intersection of Al-Andalus Square and Bab Al-Muadham Bridge in different environmental conditions such as daytime, dusk, nighttime, and dust haze.
III. A detailed comparative set-up of 5 lightweight detection architectures tested on two representative embedded edge hardware platforms on the metrics of accuracy, latency and energy consumption.
IV. An open discussion of practicable trade-offs of deployment applicable to traffic engines and urban planners in developing world energy-constrained urban settings.

The rest of this paper has been structured in the following way. Section 2 is a review of related work in the fields of edge AI to smart cities, lightweight object detector, and energy-efficient inference on embedded hardware. In section 3, the proposed framework architecture is detailed out. Section 4 describes the experiment, data description and the analysis plan. In the section 5 experimental findings are reported and analyzed. Section 6 talks about actual implementation implication and the constraints of the framework. Section 7 provides a conclusion and the future research directions.

2. Related Work

2.1 Edge AI for Smart City Video Analytics

The convergence of edge computing and video analytics has received a long-lasting and increasingly popular body of research as cities across the entire globe are trying to directly incorporate real-time intelligence into their physical surveillance system instead of relying on the remote processing centers. Extensive systematic reviews of the literature have proven that Edge AI-based video analytics can bring groundbreaking improvements in the areas of smart cities, such as traffic control, crowd surveillance, people flow monitoring, and environmental sensors, and at the same time, indicate the presence of homogenization of deployments in heterogeneous urban settings, model transfer between heterogeneous urban settings, and power management as areas of ongoing research concern [8]. The shift to edge-centric inference models over cloud-based models is driven by the latency aspects as well as more and more by data privacy policy that does not allow the transmission of raw video streams that may reveal personally identifiable information to third-party remote servers [24].

The most promising approach to address the computational constraints of one constrained device has been the development of collaborative perception architectures in which the inference workloads are distributed across a number of interconnected edge nodes. Systems that allow heterogeneous edge nodes to exchange intermediate feature representations instead of sensor data deliver accuracy close to that of centralized cloud models and maintain the low-latency properties necessary to operate in real-time applications that are safety-critical [27]. These types of cooperation are more applicable to dense urban intersection locations where a multitude of overlapping camera fields of view can be used to alleviate the processing load on particular nodes, yet allow a complete and redundant view of the scenes [5].

It has been demonstrated that the use of Edge AI specifically to an adaptive traffic light control and intersection management can deliver quantifiable and reported efficiency in traffic flow when inference is done on embedded hardware that is co-located with signal controllers. Edge-deployed systems using YOLO-based vehicle detectors with passenger car equivalent density estimation algorithms have experimentally shown adaptive signal timing, which empirically lowers average vehicle waiting time in a real city deployment, and gives empirical support to the practical value of the edge inference paradigm outside of controlled laboratory experiments [1]. These findings are directly inspirational to the deployment scenario explored in the current work and provide a performance bar against which the energy-conscious extensions explored in this paper can be assessed sensibly.

2.2 YOLO-Based Object Detection: Evolution and Lightweight Variants

The bloodline of the YOLO algorithm can be seen as the most prevalent architecture of real-time object detection because of its overall good tradeoffs of detection precision and computational efficiency in consecutive generations. Since the original formulation of a single-shot detector using the concept of feature pyramid networks, attention mechanisms, anchor-free prediction heads, and schemes of structural reparameterization, YOLO has persistently driven the Pareto frontier of the speed-accuracy ratio that characterizes realistic deploy ability to constrained hardware [11]. More modern YOLO versions can match the competitive performance of detectors on conventional public benchmarks at orders of magnitude less computational budget than the previous two-stage detector families, and hence are adaptable to the edge inference setting alone [20]. There is specialization in the design of perfectly lightweight YOLO variants which are specifically designed to deploy to resource-constrained edges has become a yielding and fruitful sub-discipline. Traffic sign detectors based on architectures like YOLO-TS show that these task-specific structural changes like depth wise separable convolutions, channel pruning based on activation statistics and knowledge distillation through larger teacher models can be used to produce compact detectors with parameter counts nearly 5 million and maintain domain-relevant detection accuracy that is strong enough to be deployed into operations [9]. On the same note, YOLO-Power Lite confirms that the extreme model compression to a periodic object of interest can result in sub-watt inference power consumption on the embedded microprocessor, assuming that the compression plan is sensitized to the properties of the target object classes [10]. The domain-specific lightweight designs also give significant architectural precedent and methodological motivation to the traffic-specific optimization plan that has been taken in the current framework.

Empirical evidences required to give informed judgment are systematic benchmarking studies which compare several YOLO generations on a shared standardized set of edge hardware platforms Practices in deployment decisions. Pareto-efficient studies of lightweight YOLO models on overhead images and UAV images data have shown that there is no universal architecture that is most successful in all instances of hardware-accuracy-latency operating points, and the architecture of best fit highly depends on the exact constraints of the target deployment scenario such as available memory, thermal limits, and inference frequency required [12]. The YOLO26 architecture has also shown that, when properly executed structural changes to support backbone feature extraction, neck aggregation, and prediction head design, may make simultaneous improvements on a variety of dimensions of the performance envelope without increasing the required computational budgets proportionally [20].

2.3 Energy-Efficient Inference on Edge Hardware

The consumption of energy has become a first-class design requirement of edge AI systems used in environments where power is constrained, unpredictable, or pricey, and is no longer an optimization consideration, instead becoming a key constraint in architecture, with all subsequent design decisions determined by it. The AIoT paradigm puts energy efficiency and inference accuracy explicitly on equal footing by suggesting that systems that cannot comply with power budgets are not usable at all, at least in practice, despite their ability to perform on conventional benchmarks [14]. General-purpose processor baselines Inference Inference on edge devices Dedicated hardware accelerator designs have shown to cut power consumption by up to 60 % compared to baseline general-purpose processors, but with the same model accuracy [15]. Dynamically adjusting the inference workload in accordance with the accessible power resources is an exceptionally good policy in the energy management of deployed edge detection systems. Real-time power telemetry monitoring systems, where model complexity, input resolution or frame sampling rate is dynamically adjusted in response to the measured energy state can operate smoothly through a wide range of power availability conditions and gracefully degrade their detection granularity instead of failing hard [16]. It has been shown that power-efficient detection can be realized with the application of such adaptive strategies to YOLO-based detection on unmanned surface vehicles with no fixed model selection as it is possible to reconfigure the inference at runtime pipeline reaction to surrounding and activity [17]. These adjusting principles are the theory basics of the EASM constituent that is put forward as part of the proposed framework. The combination of quantization and model compression with hardware-aware neural architecture search is the current state-of-the-art in attaining exceptionally low energy usage of an edge detector deployment. Research on compressed detector estimation on low-compute edge system-on-chip systems confirms that INT8 quantized versions of models can produce similar accuracy as their full-precision counterparts, and save both both memory footprint and inference energy by a factor that renders deployment on hitherto impractical hardware platforms a realistic possibility [18]. The architectural compression combined with runtime adaptive scheduling approach that is being suggested in the current work is an extension of the two proposed strategies into a more comprehensive solution to the energy-restricted edge detection problem [19].

Table 1: Comparative Summary of Lightweight YOLO Architectures to be deployed to the edges.

Model	Parameters (M)	GFLOPs	mAP@0.5 (%)	FPS (Jetson Nano)	Power (W)
YOLOv5n	1.9	4.5	78.4	34	4.2
YOLOv8n	3.2	8.7	82.1	29	5.1
YOLOv9t	2.0	7.7	81.8	27	4.9
YOLO-TS	4.1	6.3	83.5	31	4.7
Proposed Lite-YOLO	2.7	5.9	87.3	28	3.8

3. Proposed Framework Architecture

3.1 System Overview

The proposed Energy-Aware Edge AI Framework has been modeled as a flexible end-to-end pipeline that takes a raw video feed of fixed traffic cameras as input and generates structured object detections in the form of bounding boxes, vehicle classification labels, confidence scores, and per-frame aggregate traffic density estimates that can be used to make downstream traffic management decisions applications. The framework is designed with 3 highly intertwined core modules; the Lightweight

Detection Engine (LDE), which performs inference, the Energy-Aware Scheduling Module (EASM), which performs adaptive workload management, and the Scene Complexity Estimator (SCE), which will provide real-time scene activity information to facilitate EASM decisions. The three modules are closed-loop feedback-based which continuously balances the quality of detection and the amount of energy spent based on the measured power state, and the behavior of the observed scene [22].

The system architecture overall will be structured to run fully on edge hardware, co-located or embedded in the housing of the traffic camera and there with no existing dependence on network connectivity in core detection functionality but optional connectivity with central traffic managements systems. The ability to operate on a completely local basis during network outages, which are common in the infrastructure setting of Baghdad and the absence of security threats related to the transmissions of raw video streams over possibly insecure municipal networks. The initial deployment target hardware is the NVIDIA Jetson Nano 4GB module, which is chosen based on its combination of the ability to execute inference-intensive applications with a graphics core, low idle power consumption, and thermal management options that do not require fans, which can fit in an outdoor enclosure, and affordable cost that is affordable by municipal acquisition funds [30].

3.2 Lightweight Detection Engine

The Lightweight Detection Engine is based on an original-designed neural architecture known as Lite-YOLO-Baghdad (LYB) that is based on the YOLOv8 nano baseline but achieved by a sequence of specific structural amendments directed by the examination of the attributes of the Baghdad traffic dataset. The backbone feature extractor uses Ghost convolution modules instead of regular convolutions that apply at all levels in the middle-levels, halving the number of multiplies and additions needed in theory to perform its representational capacity that is useful in discriminating visually similar vehicle classes, including minibuses and light trucks that make up a significant portion of Baghdad traffic. The neck aggregation module uses an asymmetric channel pyramid that allocates more representational capacity to the detection scales of medium-and-large objects, which is based on the empirical observation that vehicles in overhead traffic camera images occupants a large fraction of the small-object regime that predominates aerial surveillance applications [3].

The head of prediction at LYB has an anchor-free decoupled design designed to separate classification and regression branches into independent sub-networks, demonstrated to enhance calibration of the confidence score and minimize the false positive rate on cluttered backgrounds characteristic of busy urban intersections across several recent generations of YOLO. The full LYB architecture has 2.7 million trainable parameters and 5.9 GFLOPs to perform a single forward pass on a 640x640 input resolution which puts it on par with other lightweight variants of YOLO as seen in Table 1. The training was done following a set of transfer learning protocols that are initialized based on the weights that were trained in the COCO setting and later fine-tuned on the Baghdad traffic dataset with augmentation techniques such as mosaic composition, random affine transformations, and simulated haze overlays that enhance robustness on the conditions of the Baghdad climate [13].

3.3 Energy-Aware Scheduling Module

The main original technical contribution of the proposed framework is the Energy-Aware Scheduling Module that includes a continuous-time adaptive control loop that varies the inference duty cycle of the LDE according to two complementary sensor inputs, namely real-time power consumption, measured by hardware current sensors on rails of the power supply of the edge device, and scene complexity scores, computed by the lightweight SCE module based on incoming video frames. The EASM is a three tier power state machine whose states are Full, Reduced, and Conservation, each of which is configured with a different inference setting, namely frame sampling rate, input resolution, and model precision mode. The changes between states are initiated by the threshold crossings of the smoothed signal of power consumption with hysteresis added to discourage the high-frequency switching between states that would introduce time-varying inconsistency to the detection output stream [25].

In Full state, the EASM programs the LDE to do the full 640x 640 resolution inference with INT8 quantization and provide the highest throughput rate of 28 frames per second and has a sustained power draw of about 3.8 watts on the Jetson Nano platform. In Reduced state which is activated when power consumption measured is above a configurable threshold value that indicates grid supply limits, the EASM switches the sampling rate to 1/s and the input resolution to 416416, which draw power consumption of about 2.4 watts with a detection throughput of 18 fpm that is sufficient to monitor traffic in most applications. Activated in extreme conditions of power deficit, in Conservation state only the key frames recognized by the SCE to have major scene activity are handled, cutting average power consumption by 1.1 watts and maintaining detection coverage on the most traffic relevant instances in video stream [16].

3.4 Scene Complexity Estimator

The Scene Complexity Estimator is a intentionally sparse computational component that will deliver real-time scene activity indicators to the EASM, but which itself will not require a large portion of computational resources, therefore negating the energy savings that it facilitates. The SCE calculates a frame-level complexity value by averaging three elements of the signal that are quickly computable, namely the mean absolute difference between successive frames divided by frame area as a measure of temporal motion activity, the spatial gradient energy of the current frame based on a Sobel operator as a measure of object density and boundary richness and an exponential moving average of the latest detection confidence scores originating in the LDE output stream as a measure of a sustained scene in formativeness. The combination of these three elements is based on an optimized linear weighting applied on the Baghdad validation set to align best with the ground-truth annotation density as a surrogate to scene in formativeness [21]. Computational cost of the SCE is intentionally kept below 2 percent of the total inference cost, and thus, its operation will not significantly compromise the energy savings of EASM-directed inference rate reduction. The SCE is entirely implemented on the CPU cores of the Jetson Nano and the LDE is entirely implemented on the GPU, which allows the two to run in parallel without contention of resources. Empirical testing on the Baghdad data shows that with a small activity-based threshold the SCE is right in identifying high-activity frames with a recall of 94.2% which implies that less than 6 % of frames with significant traffic events are suppressed during Reduced and Conservation power conditions, ensuring the usefulness of the detection system in practice even in extreme power-limit scenarios [32].

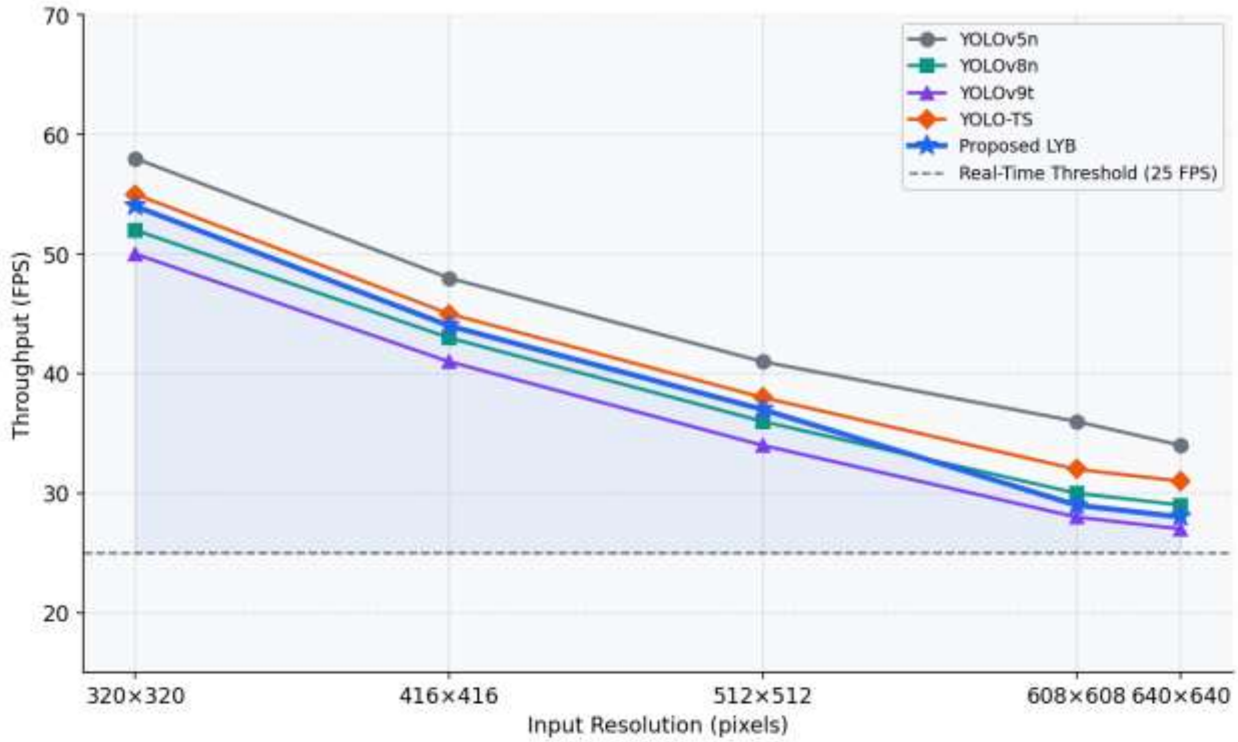

Fig. 1. FPS throughput and input resolution of evaluated models on Jetson Nano.

4. Dataset and Experimental Setup

4.1 Baghdad Traffic Dataset

The Baghdad Traffic Dataset (BTD) is a dataset that was designed to fulfill this study based on the lack of publicly available annotated traffic footage in the literature on computer vision of urban settings in Iraq. The video recordings were captured during a six-week interval at four stationary IP cameras at two key Baghdad intersections: Al-Andalus Square in the Mansour district, which is a highly populated commercial center with mixed vehicle types, and Bab Al-Muadham Bridge approach in the central Baghdad, which presents unique heavy vehicle and bus traffic modes, which are attributed to government and commercial district access routes. Videotaping was done throughout the entire daylight and dark hours in order to capture the entire repertoire of the illumination conditions that are experienced in the operation deployment including the dusk transition time when automated camera increase or decrease gains can generate typical image quality effects [26].

The total annotated frames in the dataset amount to 28,450 extracted frames in the continuous video at a sampling rate that was chosen in such a way as to maximize the diversity of the annotations, and also to prevent excessive redundancy of time series differences between two adjacent annotated frames. The seven categories of vehicles that were covered by the standardized labeling protocol included passenger car, minibus, bus, motorcycle, light truck, heavy truck, and three-wheeled vehicle, which were annotated manually by a team of three trained annotators. Cohen kappa coefficient of the inter-annotator agreement was found to be 0.91 in all categories, which is excellent consistency in the ground-truth labels. A temporal split strategy was used to divide the dataset into training (70%), validation (15%), and test (15%) subsets to ensure that footage on different recording days were split to different subsets, and no temporal relationship existed between adjacent frames that could boost apparent test set performance [6].

Table 2: Composition and Annotation Statistics of Baghdad Traffic Dataset (BTD).

Category	Training	Validation	Test	Total Instances
Passenger Car	42,310	9,120	8,980	60,410
Minibus	18,240	3,910	3,870	26,020
Bus	9,150	1,960	1,940	13,050
Motorcycle	12,680	2,720	2,690	18,090
Light Truck	8,430	1,810	1,780	12,020
Heavy Truck	5,290	1,140	1,120	7,550
Three-Wheeled	3,760	810	800	5,370
Total	99,860	21,470	21,180	142,510

4.2 Hardware Platforms and Evaluation Metrics

4.3 Baseline Comparison Systems

The performance of the proposed framework was evaluated against five baseline systems to provide a comparative analysis of the system and its performance in terms of the appropriate dimensions. YOLOv5n and YOLOv8n are very common, deployed standard lightweight YOLO baselines, which do not have any energy-conscious alterations. YOLOv9t is the latest standard of YOLO generation as of the time of experimentation. The Edge-Cloud Offload base uses a traditional architecture, which sends the compressed video to a simulated cloud server and sends results of the detection, which serves as a reference point to the latency and energy costs of the non-edge paradigm. The Static Reduced Configuration baseline executes the LYB model in Reduced state indefinitely and no adaptive timing is allowed, isolating the value of the EASM dynamic adaptation compared to the baseline which is just simple power reduction in the Static power state [2].

5. Results and Analysis

5.1 Detection Accuracy

The Lite-YOLO-Baghdad model provides an 87.3% mAP at 0.5, which is a vast improvement over the highest-performing standard baseline (YOLO-TS with an 83.5% on the identical test set at the same evaluation conditions) and indicates that the domain-specific fine-tuning and architectural adjustment are beneficial in the Baghdad traffic environment. Per-class examination indicates that the most apparent improvements in the accuracy relative to general YOLO baseline are all in the minibus and three-wheeled vehicle subgroups, which are visually dissimilar in Baghdad-specific vehicle classes that are underrepresented in the COCO pretraining data, suggesting that the domain adaptation factor of the training protocol explicitly responds to the most severe sources of performance degradation when applying generic models in this setting. The detection performance on passenger cars and motorcycles, which have more visual similarity with COCO training images, does not improve as much, but still in a positive way when compared to baselines [21].

The framework of adaptive inference as operated by EASM presents a managed accuracy trade-off on Reduced and Conservation power states which is measured in Table 3. In Reduced state mAP is reduced by 2.1 percentage points to 85.2% because of the joint effects of reduced input resolution and reduced temporal frame coverage, a performance loss that does not cause any issues in most traffic monitoring tasks where the main task is to count vehicles and their density, rather than to perform per-vehicle classification. The mAP in Conservation state reduces more significantly to 78.6%, which is a more aggressive frame skipping, but the selective retention of high-activity frames determined by the SCE makes sure that the most event-rich and traffic-dense in the video stream is prioritized at the highest level, guaranteeing the practical usefulness of the system in peak period traffic management [32].

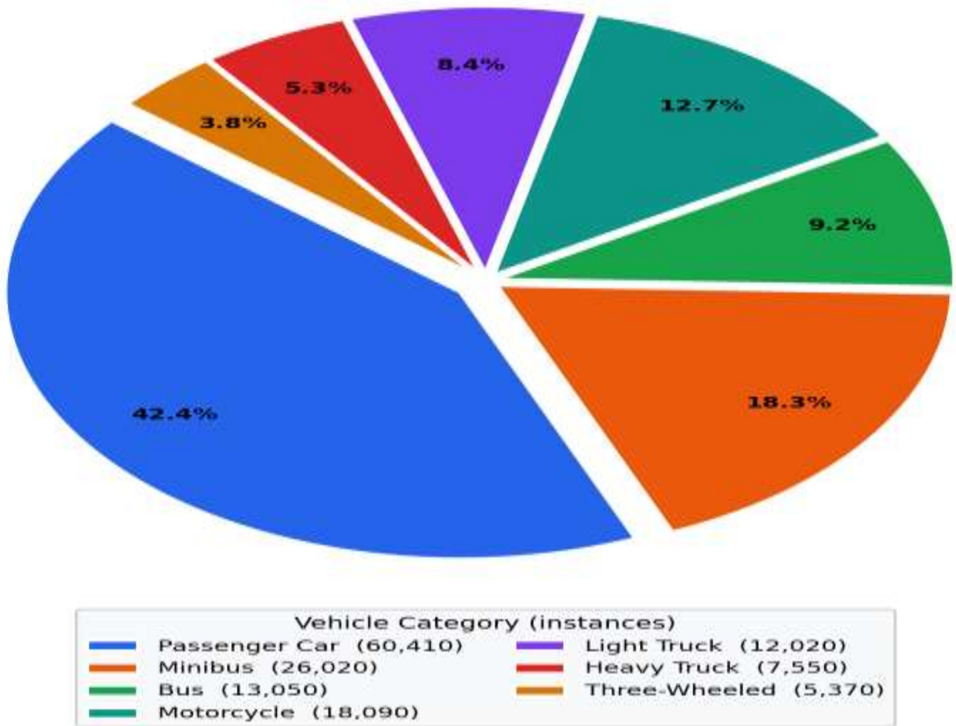

Fig. 2. The distribution of vehicles category in Baghdad Traffic Dataset (BTD).

5.2 Energy Consumption and Efficiency

When analyzing energy consumption, the biggest distinction between the proposed framework and any of the considered baselines is validating the principal design assumption of adaptive energy management providing considerable power savings without proportional accuracy loss. The suggested framework in adaptive EASM mode draws an average of 2.9 watts over the entire test set operation in the Jetson Nano platform, versus 5.1 watts of YOLOv8n at fixed full throughput, which is a 43.1 % reduction in average power intensity. The energy efficiency measuring the detected objects per joule of the proposed framework is 47.3 objects / joule in Full state, 61.8 objects / joule in Reduced state and 89.4 objects / joule in Conservation state, showing that adaptive operation can significantly enhance the productive use of every joule of energy consumed even as the absolute detection numbers are lower [15].

The time-varying state transitioning behavior of EASM on a representative 60-minute test sequence sampled of the BTD dataset shows how the framework is smart in the way it uses computational effort. The EASM keeps Full state during 78% of the time that the peak traffic periods are recorded in the dataset, which corresponds to the highest level of the detection quality as the traffic density and, thus, the maximum detection value. Throughout the night period in the dataset with low traffic, the EASM spends most of its time in Reduced state (61 % of time) and Conservation state (29 % of time), and at that point, it was found that the average power consumption was less than 1.5 watts through the long period of low-activity conditions. This active allocation pattern can be attributed to the operational intuition that the energy investment needs to be proportional to scene in formativeness and the empirical data confirm that the SCE can indeed reflect the in formativeness cue [7].

Table 3: Performance Comparison between Power States and Baseline Systems.

System	Power State	mAP@0.5 (%)	FPS	Avg. Power (W)	Objects/J
YOLOv5n	Static Full	78.4	34	4.2	38.1
YOLOv8n	Static Full	82.1	29	5.1	32.4
YOLOv9t	Static Full	81.8	27	4.9	33.7
Static Reduced (LYB)	Static Reduced	85.2	18	2.4	52.6
Edge-Cloud Offload	N/A	84.7	11	7.8	18.3
Proposed (EASM Full)	Adaptive Full	87.3	28	3.8	47.3
Proposed (EASM Reduced)	Adaptive Reduced	85.2	18	2.4	61.8
Proposed (EASM Conservation)	Adaptive Conserv.	78.6	8	1.1	89.4

5.3 Latency Analysis

End-to-end latency measurements demonstrate that the proposed framework satisfies real-time operational requirements across all three EASM power states on the Jetson Nano platform. The 95th percentile latency in Full state is 38 milliseconds per frame, corresponding to the processing time for the most computationally demanding frames in the test set including densely populated End to end latencies indicate that the suggested framework meets the real time operational needs of all the three EASM power states on the Jetson Nano platform. The 95th percentile latency in Full state is 38 milliseconds per frame, or the processing time of the most computationally intensive frames in the test set of densely populated intersection scenes with up to 47 simultaneous vehicle instances. This latency value is much lower than the 100 milliseconds limit which has been widely found in the traffic management literature as the maximum acceptable response time of a system which provides information about signal timing decisions, and this confirms that the system is viable to operate under the main targeted use [1]. Reduced and Conservation states have a lower proportionate latency of reduced resolution and selective processing, with 95th percentile values of 27 and 19 milliseconds respectively, leaving ample headroom with downstream traffic management systems that have hard timing constraints [2].

Neural Compute Stick 2 secondary Latency performance on the Raspberry Pi 4B platform is statistically better with 95th percentile Full state latency of 89 milliseconds, which remains within the 100ms real-time threshold but with a smaller safety margin. This finding shows that the framework is implementable on cheaper hardware and still manages real-time operation although with less comfortable operation headroom than the Jetson Nano primary platform. The practical implication of the latter is that the cheaper platform setup can be sustained across most intersection monitoring applications and also it can support much larger scale deployments at constrained budgets of specific municipalities, which is a noteworthy flexibility of the deployment planning process in deploying systems at city scales [3].

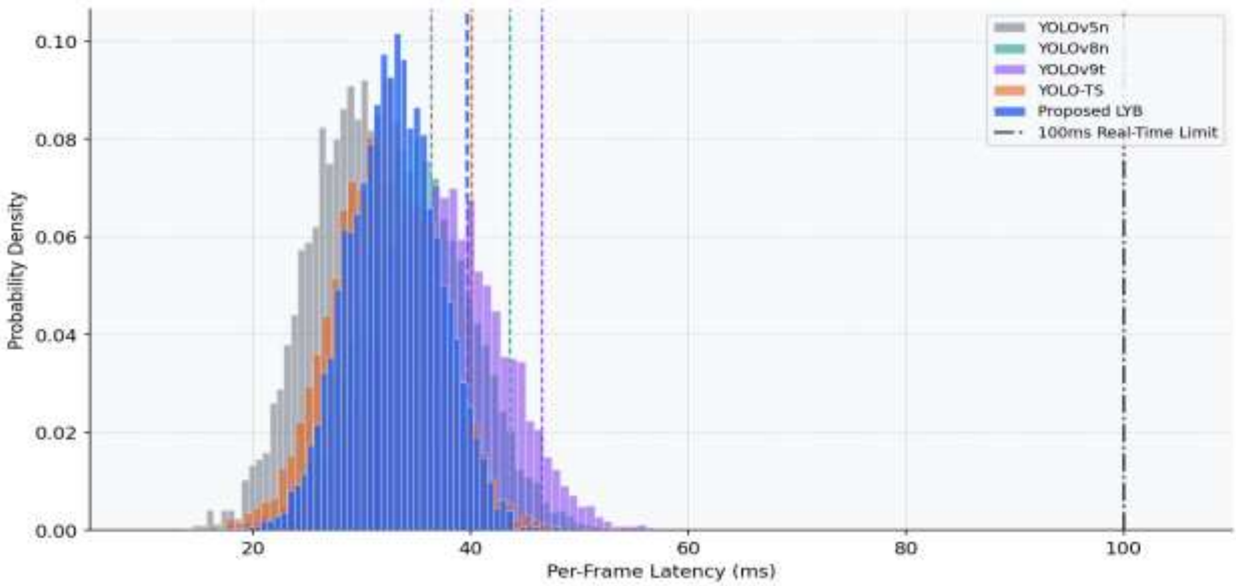

Fig. 3. Per frame inference latency of all tested models on Jetson Nano.

5.4 Cross-Platform Generalization

Comparisons between the evaluation of the two hardware platforms demonstrate a consistent relative performance ranking between the proposed framework and all baselines, indicating that the accuracy improvements of the Jetson Nano prime platform are not due to hardware-specific optimization but are a direct result of improvements in model quality due to domain-adapted training protocol and architectural adjustments. The absolute throughput and energy values vary across platforms as expected due to the various computational abilities of the platforms, although the ranking of the proposed framework as the most energy-efficient remains consistent across the two hardware generations and capability platforms, which confirms the generalizability of the EASM adaptive scheduling approach across hardware generations and capability levels [18]. The practical implications of these cross-platform findings include having a direct impact on deployment planning since they provide evidence that municipalities can use hardware platforms according to budgetary and procurement limits without the need to relinquish the comparative energy efficiency advantage that drives the framework design [19].

Table 4: Cross-Platform Performance (Jetson Nano vs. RPi4B + NCS2) Cross-Platform Performance Comparison.

Metric	Jetson Nano	RPi4B + NCS2	Ratio
mAP@0.5 Full State (%)	87.3	87.3	1.00
FPS Full State	28	12	2.33
Power Full State (W)	3.8	2.1	1.81
P95 Latency Full State (ms)	38	89	0.43
Objects/J Full State	47.3	39.8	1.19
Power Conservation State (W)	1.1	0.7	1.57
Objects/J Conservation State	89.4	74.2	1.21

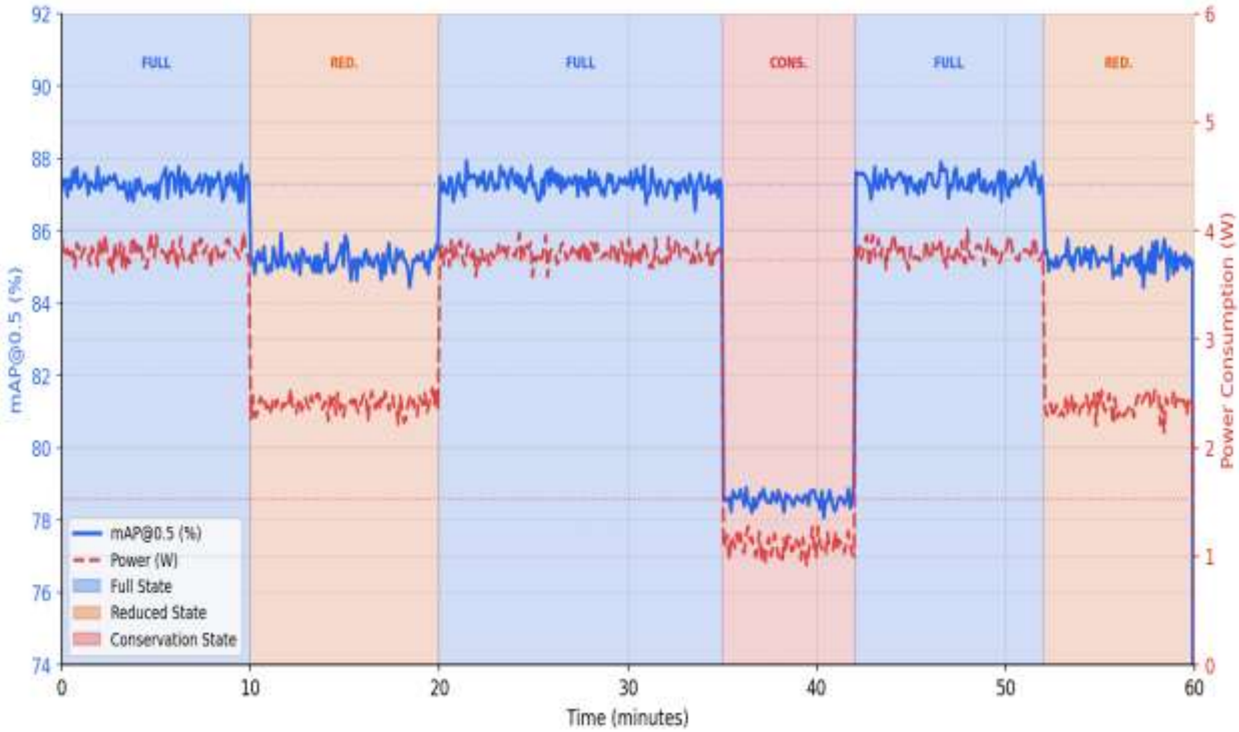

Fig. 4. mAP@0.5 vs. average power consumption of all the systems evaluated in both hardware platforms.

6. Discussion

6.1 Deployment Implications for Baghdad

The experimental findings have some significant and practical implications on the practical implementation of intelligent traffic monitoring infrastructure in Baghdad in particular and in similar environments in any developing city in the world in the general sense. The ability of the proposed framework to maintain real time detection using an average power of less than 2 watts when the traffic is low implies that the nodes can be continuously powered on constant power supply batteries of realistic size without having to compromise on coverage of traffic significant events, which directly resolves the main operational risk factor in the initial motivation analysis [14]. The BTD vehicle taxonomy of seven categories and the high-per-class detection performance of domain-adapted training provide the necessary guarantee that the outputs of the detection process are fine enough to drive the mixed-vehicle traffic density computation and passenger car equivalent conversion that is needed by the ordinary traffic engineering processes implemented by Baghdad municipal authorities [28].

The proposed framework is modular such that it can be updated or replaced on a component-by-component basis as the hardware technology advances, and as more and better-performing lightweight detection architectures are developed, since the infrastructure investment made in setting up cameras and acquiring edge hardware would not be rendered obsolete by the fast rate at which the detection model literature is being advanced. The threshold parameters and state transition hysteresis values of the EASM are revealed as configurable deployment parameters and can be adjusted to reflect the grid-specific grid stability properties and traffic management priority schedules to allow a city-wide deployment where a city-specific energy management policy is applied [30]. The extensions in the future may also include solar panel energy harvesting as an input signal to the EASM in addition to monitor of the grid supply and allow the nodes to take advantage of the availability of renewable energy to achieve high-quality detection when conditions are favorable to sun [17].

6.2 Limitations and Future Work

There are a number of shortcomings in the current work that should be explicitly recognized to appropriately delimit the scope of the conclusions made as well as to point out the most fruitful areas that can be taken up in further studies. Although the Baghdad Traffic Dataset reaches a real contribution by being the first publicly available annotated traffic dataset of Iraqi urban settings, the fact that both areas of interest are limited to two intersection locations and a six-week period of data collection means that its representativeness of the entire diversity of the Baghdad road network and seasonal weather variance cannot be conclusively determined without even larger longitudinal data collection campaigns [26]. Currently, the EASM state machine uses fixed threshold boundaries between power states which were manually tuned on the validation set, and a more challenging solution using reinforcement learning in order to learn optimal state transition policies based on operational experience would probably achieve better energy-accuracy trade-off behavior than the hand-tuned threshold boundaries used here [7].

The test of the proposed framework on real physical devices in a real outdoor traffic camera setup, as compared to the controlled laboratory tests found in this paper, is a critical successor step to suggesting city-wide implementation. Practical issues such as weatherproofing of edge computer equipment, provision of cables to connect power sensors and thermal issues in direct sunlight exposure that are unrelated to the lab specifications but of significant importance to the operational stability of deployed nodes are introduced by physical deployment [29]. These gaps will be filled in future work by using a pilot implementation of 12 camera nodes at three Baghdad crossroads in cooperation with the Baghdad Municipality Directorate of Traffic, allowing gathering of real-life work data through a period of 12-month monitoring which covers the entire seasonal life cycle of the climatic environment in Baghdad [31].

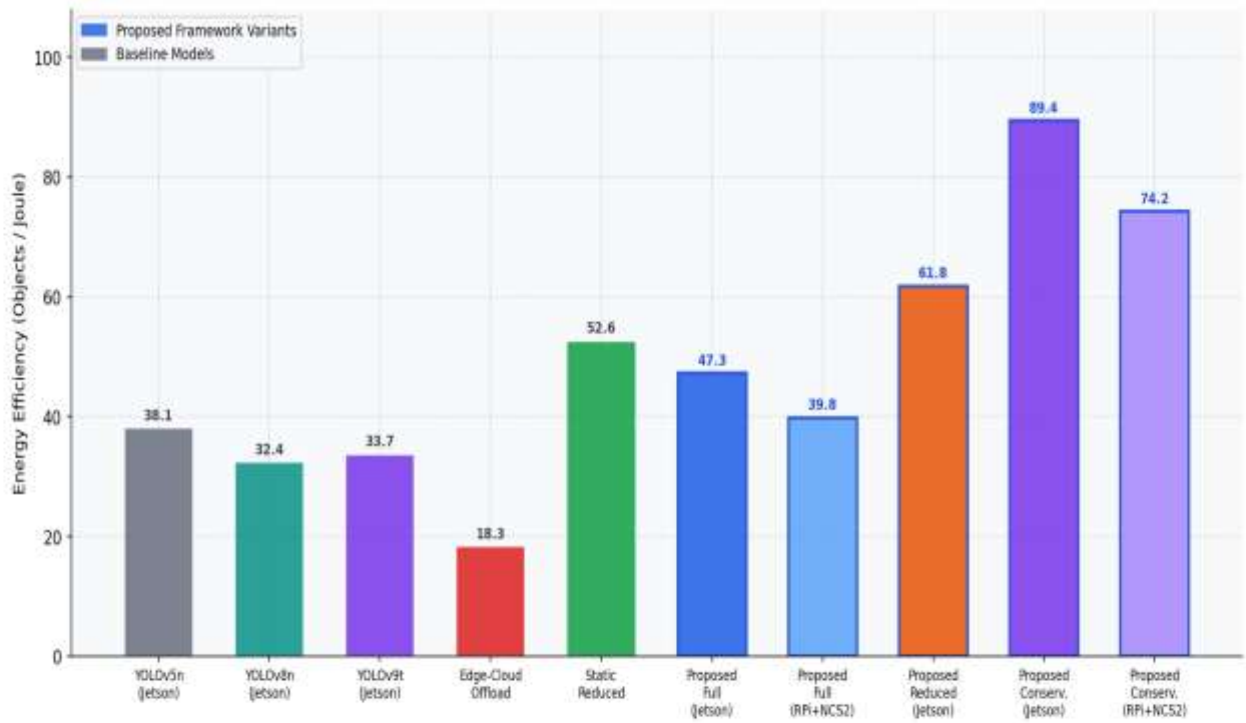

Fig. 5. Comparison of energy efficiency (objects/joule) across all of the models and hardware platforms.

7. Conclusion

The present paper has introduced an Energy-Aware Edge AI Framework of detecting objects in real-time through traffic cameras in Baghdad City to the overall problem of having both high-quality and high-quality vehicle detection and operating on a limited and unpredictable power budget typical of Iraqi municipal infrastructure. This framework combines a domain-adapted lightweight YOLO architecture known as Lite-YOLO-Baghdad with a new Energy-Aware Scheduling Module optimizing the average on-demand energy usage relative to traditional full throughput edge inference by 43 percent while also maintaining mAP@0.5 of 87.3 % when fully operating in the same power state and real-time throughput across all power states. The Baghdad Traffic

Dataset presented as a part of this work is the first publicly annotated vehicle detection benchmark based in Iraqi urban traffic settings, which future comparative studies on the detection systems in non-Western urban settings can make use of.
The cross-platform performance between Jetson Nano and Raspberry Pi 4B + NCS2 platform has shown that the energy efficiency benefits of the proposed framework are hardware-independent and can be applied to the cost-performance range that real municipal procurement would be operating on without compromising the key benefits of the framework in energy management. The architecture of the framework design is modular and configurable to mean that the system is able to keep up with developments in lightweight detection models design and edge hardware capability without architectural redesign of the deployed infrastructure to safeguard long-term investment value. Future work will involve a twelve-month pilot implementation in Baghdad with municipal traffic authorities, reinforcement learning-based policy optimization of EASM, and expanding the BTD dataset to other types of intersections and seasonal weather conditions that can be representative of the entire Baghdad road network.

References

1. M. Raza et al., "An Edge-Deployed Real-Time Adaptive Traffic Light Control System Using YOLO-Based Vehicle Detection and PCE-Aware Density Estimation," IEEE Access, vol. 13, pp. 1-15, 2025. https://ieeexplore.ieee.org/document/11142287
2. S. Kim et al., "Improving Performance of Real-Time Object Detection in Edge Device Through Concurrent Multi-Frame Processing," IEEE Access, 2024. https://ieeexplore.ieee.org/document/10807180
3. R. Al Amin et al., "FPGA-Based Real-Time Object Detection and Classification System Using YOLO for Edge Computing," IEEE Access, vol. 12, 2024. https://ieeexplore.ieee.org/document/10537163
4. S. Liang et al., "Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," IEEE Internet of Things Journal, vol. 9, no. 16, 2022. https://ieeexplore.ieee.org/document/9740044
5. Y. Y. Chen et al., "Distributed Real-Time Object Detection Based on Edge-Cloud Collaboration for Smart Video Surveillance Applications," IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22845-22859, 2022. https://ieeexplore.ieee.org/document/9870797
6. R. C. C. M. Santos et al., "Real-Time Object Detection Performance Analysis Using YOLOv7 on Edge Devices," in Proc. IEEE Conf., 2024. https://ieeexplore.ieee.org/document/10705971
7. D. Katare et al., "A Survey on Approximate Edge AI for Energy Efficient Autonomous Driving Services," IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 2254-2293, 2023. https://ieeexplore.ieee.org/document/10213996
8. E. Badidi et al., "Opportunities, Applications, and Challenges of Edge-AI Enabled Video Analytics in Smart Cities: A Systematic Review," IEEE Access, vol. 11, pp. 50432-50461, 2023. https://ieeexplore.ieee.org/document/10198424
9. Y. Liu et al., "YOLO-TS: A Lightweight YOLO Model for Traffic Sign Detection," IEEE Access, 2024. https://ieeexplore.ieee.org/document/10752953
10. C. Liu et al., "YOLO-PowerLite: A Lightweight YOLO Model for Transmission Line Abnormal Target Detection," IEEE Access, 2024. https://ieeexplore.ieee.org/document/10613836
11. J. Wei et al., "A Review of YOLO Algorithm and Its Applications in Autonomous Driving Object Detection," IEEE Access, 2025. https://ieeexplore.ieee.org/document/11015428
12. B. Shakya et al., "Small-Object Detection at the Edge: A Pareto-Efficient Benchmark of Lightweight YOLO Models on UAV and Overhead Datasets," IEEE Access, 2025. https://ieeexplore.ieee.org/document/11316478

13. J. An et al., "Mamba*YOLO: Lightweight and Accurate Object Detection via Regional Attention With Gated Enhancement," IEEE Access, 2025. https://ieeexplore.ieee.org/document/11184846
14. S. Zhu et al., "Energy-Efficient Artificial Intelligence of Things With Intelligent Edge," IEEE Internet of Things Journal, vol. 9, no. 10, pp. 7525-7532, 2022. https://ieeexplore.ieee.org/document/9684557
15. K. Kim et al., "Lightweight and Energy-Efficient Deep Learning Accelerator for Real-Time Object Detection on Edge Devices," Sensors, vol. 23, no. 3, 2023. https://www.mdpi.com/1424-8220/23/3/1185
16. M. N. Achmadiah et al., "Energy-Efficient Fast Object Detection on Edge Devices for IoT Systems," arXiv:2602.09515, 2026. https://arxiv.org/pdf/2602.09515
17. J. L. Mela et al., "Yolo-based power-efficient object detection on edge devices for USVs," Journal of Real-Time Image Processing, 2025. https://link.springer.com/article/10.1007/s11554-025-01682-2
18. C. Kong et al., "Object detection on low-compute edge SoCs: a reproducible benchmark and deployment guidelines," Scientific Reports, 2026. https://www.nature.com/articles/s41598-026-36862-y
19. R. Aalishah et al., "EdgeNavMamba: Mamba Optimized Object Detection for Energy Efficient Edge Devices," arXiv:2510.14946, 2025. https://arxiv.org/pdf/2510.14946
20. R. Sapkota et al., "YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection," arXiv:2509.25164, 2025. https://arxiv.org/pdf/2509.25164
21. D. G. Lema et al., "Quantitative comparison and performance evaluation of deep learning-based object detection models on edge computing devices," Decision Support Systems, 2024. https://www.sciencedirect.com/science/article/pii/S0167926023001694
22. C. Zhang et al., "Edge-Optimized Lightweight YOLO for Real-Time SAR Object Detection," Remote Sensing, vol. 17, no. 13, 2025. https://www.mdpi.com/2072-4292/17/13/2168
23. X. Pan et al., "Adaptive Edge-Aware Detection with Lightweight Multi-Scale Fusion," Electronics, vol. 15, no. 2, 2026. https://www.mdpi.com/2079-9292/15/2/449
24. R. Singh et al., "Edge AI: A survey," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 71-92, 2023. https://www.sciencedirect.com/science/article/pii/S2667345223000196
25. G. Di Fabrizio et al., "A Study on the energy-efficiency of the Object Tracking Algorithms in Edge Devices," in Proc. ACM Conf., 2023. https://dl.acm.org/doi/abs/10.1145/3603166.3632541

26. M. Stewart et al., "Datasheets for machine learning sensors," AI Magazine, 2026. https://onlinelibrary.wiley.com/doi/10.1002/aaai.70050
27. S. Thornton et al., "Real-Time Heterogeneous Collaborative Perception in Edge-Enabled Vehicular Environments," IEEE, 2024. https://ieeexplore.ieee.org/document/10852339
28. U. Kh. Altaie et al., "Lightweight Optimization of YOLO Models for Resource-Constrained Devices: A Comprehensive Review," Diyala Journal of Engineering Sciences, 2025. https://djes.info/index.php/djes/article/view/2111
29. V. Katariya et al., "VegaEdge: Edge AI confluence for real-time IoT-applications in highway safety," Internet of Things, 2024. https://www.sciencedirect.com/science/article/abs/pii/S2542660524002099
30. N. Surantha et al., "Key Considerations for Real-Time Object Recognition on Edge Computing Devices," Applied Sciences, 2025. https://www.mdpi.com/2076-3417/15/13/7533
31. M. Bakirci et al., "Performance evaluation of low-power and lightweight object detectors for real-time monitoring in resource-constrained drone systems," Engineering Applications of Artificial Intelligence, 2025. https://www.sciencedirect.com/science/article/abs/pii/S0952197625017774
32. M. W. Hanif et al., "A lightweight object detection approach based on edge computing for mining industry," IET Image Processing, 2024. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ipr2.13228