

Article

Energy-Aware Edge Vision for Event-Level Fire Detection with YOLO-Equipped UAVs

Francisco-Jose Alvarado-Alcon ^{1,2} , Rafael Asorey-Cacheda ^{1,2,*} , Joan Garcia-Haro ^{1,2} 
and Antonio-Javier Garcia-Sanchez ^{1,2} 

- ¹ Department of Information and Communication Technologies, Universidad Politécnica de Cartagena Member of European University of Technology EUT+, 30202 Cartagena, Spain; francisjose.alvarado@upct.es (F.-J.A.-A.); joang.haro@upct.es (J.G.-H.); antoniojavier.garcia@upct.es (A.-J.G.-S.)
- ² People Oriented Smart Technologies Laboratory, European Research Group, European University of Technology, European Union, 10000 Troyes, France
- * Correspondence: rafael.asorey@upct.es

Highlights

What are the main findings?

- Joint optimization of the YOLO scale, confidence threshold, and inference stride reduces the energy cost of onboard UAV wildfire detection by up to 4.4 times.
- Compact YOLOv5 variants outperform larger models in the accuracy–energy trade-off on the evaluated aerial fire dataset.

What are the implications of the main findings?

- Event-driven onboard vision can extend the UAV wildfire-monitoring endurance by reducing the inference and transmission energy.
- Temporal subsampling enables longer surveillance missions while preserving near-perfect event-level fire detection for persistent events.

Abstract

Unmanned aerial vehicles (UAVs) are increasingly being used for early wildfire monitoring in remote areas, but UAV endurance is fundamentally constrained by the battery capacity. This work presents an energy-aware edge-vision framework for UAV fire detection that jointly models neural inference and wireless communication and optimizes the operating point of the complete onboard pipeline. Five you only look once (YOLO)v5 scales were fine-tuned on a YOLO-formatted version of the FLAME aerial fire dataset, which was extended with multi-frame fire tracking to enable event-level evaluations. We jointly optimized the model scale, detection confidence threshold, and inference stride using theoretical and empirical estimators that balance energy consumption against the probability of detecting fire events. The results showed that compact YOLOv5 models provide the best trade-off between energy and accuracy for this UAV application: larger variants increase the inference cost without consistent recall gains on the evaluated dataset. In addition, temporal subsampling reduces the total energy approximately in proportion to the stride while preserving near-perfect event-level detection for fires of a moderate duration. The optimized configuration lowers energy consumption by up to 4.4 times with only a 0.03% reduction in recall, supporting longer-endurance UAV missions for wildfire monitoring.

Keywords: edge artificial intelligence; energy optimization; object detection; onboard vision; unmanned aerial vehicle; wildfire detection



Academic Editor: Giordano Teza

Received: 15 June 2026

Revised: 29 June 2026

Accepted: 9 July 2026

Published: 11 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Wildfires represent one of the most severe and rapidly escalating natural hazards worldwide, with an increasing frequency, intensity, and spatial extent driven by climate change and land-use dynamics. Early detection is a critical factor in mitigating the wildfire impact, as the probability of successful containment decreases dramatically with detection delay. In this context, unmanned aerial vehicles (UAVs) equipped with onboard vision-based intelligence have emerged as a promising solution for scalable, flexible, and high-resolution wildfire monitoring, particularly in remote or difficult-to-access environments [1].

Recent advances in deep learning-based object detection, and in particular, the you only look once (YOLO) family of convolutional neural networks (CNNs), have enabled real-time fire and smoke detection with a high accuracy from aerial imagery. YOLO-based detectors are especially attractive for UAV deployment due to their favorable trade-off between the detection performance and the computational complexity [2–4]. However, deploying such models onboard drones introduces stringent energy constraints, as UAV endurance is fundamentally limited by the battery capacity. As a result, the continuous execution of deep neural network inference, together with the wireless transmission of detection results, can significantly reduce the mission duration and operational effectiveness.

While prior work has extensively explored the accuracy and robustness of vision-based fire-detection models, energy consumption has received comparatively limited attention, particularly from a system-level perspective. In practice, the energy cost of an autonomous fire-detection pipeline is not dominated by inference alone, but rather by the joint contribution of onboard processing, sensing, and communication. Moreover, these components are strongly interdependent: increasing the inference frequency improves the detection latency and recall, but raises the computational energy consumption, while lower confidence thresholds may increase the number of transmitted alerts, thus raising the communication energy costs.

Motivated by these challenges, this work addresses the problem of energy optimization for UAV-based fire-detection systems using deep learning. Building on previous experimental studies [5,6], we modeled the energy consumption associated with both YOLO-based inference and transmitting detection results to a ground station. Five variants of the YOLOv5 architecture (from nano to extra-large) [7] were trained and evaluated for fire detection on the FLAME dataset [2,8], which consists of aerial video footage acquired by a UAV during prescribed burns. This enables a detailed analysis of the trade-offs between the model complexity, detection performance, and energy usage under real UAV imaging conditions. To support an event-level recall analysis, we extended the dataset with fire-tracking annotations and released this extension in [9].

Unlike approaches that focus solely on model compression or hardware acceleration, we adopted a holistic optimization framework that jointly considers (i) the YOLO model size, (ii) the detection confidence threshold, and (iii) the inference execution frequency. By explicitly modeling how these parameters affect both the inference energy and communication energy, we identified operating points that minimize the total energy consumption while preserving reliable and timely fire detection. This formulation is particularly relevant for long-endurance UAV missions, where energy savings directly translate into an increased coverage area and monitoring time.

This paper makes four main contributions:

- An energy consumption model for UAV-based fire detection that jointly accounts for deep learning inference and result transmission.
- A comparative evaluation of the accuracy–energy trade-offs of five YOLOv5 models specifically trained for fire detection and validated on real UAV-acquired aerial footage.

- A multi-parameter optimization strategy that identifies energy-efficient configurations under different operational constraints.
- An extension of the FLAME dataset with fire-tracking annotations to support event-level recall analyses over time.

By providing quantitative insights into the interaction between the detection performance and the energy expenditure, this work contributes to the design of sustainable, intelligent UAV systems for wildfire monitoring and complements recent efforts toward green artificial intelligence (AI) and energy-aware edge intelligence. In Section 2, we introduce some relevant works regarding fire detection with UAVs and edge-computing optimization. Section 3 describes the used methodology, and Section 4 presents the results of this work. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Fire Detection with UAVs

The mobility, low flight altitude, and rapid deployability of UAVs make them compelling platforms for early wildfire detection and situational assessments [10,11]. Progress, however, is hindered by the scarcity of large, well-annotated datasets that capture diverse fire appearances and backgrounds, limiting both training and benchmarking [10,12].

A common starting point due to its simplicity and easier annotation workflow is image classification, where an entire frame is assigned a single label. For example, the authors of [10] trained deep models to classify images into four categories (flame with smoke, flame without smoke, smoke without flame, and neither) and compared RGB (images where each pixel is defined by the amount of red, green, and blue colors), infrared (IR), and RGB+IR inputs; their ablation showed that RGB contributed most to the final accuracy. In a similar line, the work by [11] combined a MobileNet-v2 [13] feature extractor with a recurrent module for fire/no-fire classification.

Object detection localizes and classifies each fire instance, and therefore requires bounding box annotations, which are more labor-intensive, but enable downstream localization and selective transmission [7,10]. A representative approach is shown in [14], where RGB imagery captured by a UAV was processed onboard with YOLOv3; upon detection, the system shared the image and GPS coordinates with a ground station. More recently, Luan et al. [12] redesigned a compact YOLOv8n variant by introducing attention, a lighter backbone, and weighted feature fusion for fire detection. Although tailored for low-computational deployments, that work did not explicitly quantify the impact of inference and wireless transmission on energy.

Segmentation methods predict a label per pixel, yielding precise fire contours at the cost of even more complex annotations. For instance, Yandouzi et al. [15] explored Mask R-CNN and YOLO-based segmentation under a fog-computing setup, where UAVs continuously stream video and inference runs offboard. While this may simplify airborne software, continuous streaming can be energetically costly for battery-constrained platforms; in contrast, onboard detection allows for the event-driven transmission of compact regions of interest.

Avoiding onboard computing and instead transmitting sensor data to ground servers for processing may be appropriate when relying primarily on low-bandwidth telemetry (temperature, humidity, smoke, etc.), as in [16]. When images are the primary signal, however, continuous transmission can markedly shorten the flight endurance. Even with onboard inference, we advocate transmitting the positive detections (cropped evidence with metadata) so that ground operators can visually confirm the presence and operational significance of a suspected fire, while keeping communications overhead low.

Overall, prior UAV-based fire-detection studies often adopt edge computing without modeling how the interplay between onboard inference and communications affects the UAV's energy budget and, consequently, the flight time. Our work does not propose a new architecture; instead, we fine-tuned established YOLOv5 detectors for object detection and focused on modeling and quantifying the energy costs of both inference and result transmission.

We adopted object detection as a deliberate system-design choice aligned with the operational workflow. Our pipeline transmits to the ground station a cropped region of interest around the detected fire for confirmation, either by a human operator or by a more accurate downstream model. With image-level classification, the detector provides no localization; therefore, the full frame must be transmitted to support verification, increasing the communication cost. Conversely, segmentation provides pixel-level contours that are unnecessary for this purpose: the transmitted evidence must ultimately be stored and displayed in standard rectangular image formats, and a tight bounding box is sufficient to produce compact, interpretable crops.

Thus, object detection best matches our objectives by enabling selective transmission with a limited bandwidth and energy overhead while preserving actionable spatial context. Therefore, the previously discussed scarcity of datasets is particularly relevant for the present work, because the proposed event-level evaluation requires not only aerial fire imagery, but also temporally ordered UAV frames in which the same fire instances remain visible across multiple frames. Among the publicly available datasets considered, FLAME was the only dataset we found that simultaneously provides UAV-acquired aerial fire imagery, pixel-level fire annotations that can be converted into bounding boxes, and sequential frames from the same video, thereby enabling multi-frame fire tracking and event-level recall analyses.

Among the numerous object-detection architectures, we selected YOLOv5 for both methodological and deployment-oriented reasons. YOLO-family detectors are one-stage architectures designed for real-time object detection, which makes them well suited to onboard UAV perception, where the inference latency, memory footprint, and energy consumption are constrained. In particular, YOLOv5 provides a mature and reproducible family of model scales, from nano to extra-large, implemented within a common training and inference framework. This is important for the present study because it allows the effect of the model capacity on the detection performance and energy consumption to be analyzed while keeping the detector family, input resolution, and training procedure consistent. Previous fire-detection studies also support the use of YOLOv5 as a strong baseline. For example, Casas et al. [4] reported that YOLOv5 achieved a favorable performance compared with YOLOv8 in wildfire- and smoke-detection scenarios in terms of the detection metrics, computational complexity, and test time. Similarly, Uddin et al. [17] compared YOLOv5, YOLOv8, and other CNN-based fire-detection architectures, showing that YOLO-based detectors outperformed the remaining tested approaches, although the best-performing YOLO version depended on the model scale and dataset considered. Therefore, YOLOv5 was selected as a mature, scalable, and energy-relevant architecture family that is well aligned with the objective of evaluating accuracy–energy trade-offs in UAV-based fire detection.

2.2. Edge-Computing Optimization

Edge computing executes computations close to where the data are generated, rather than sending all raw data to centralized servers. In the context of UAVs, this typically means running perception and decision modules on the aircraft or on nearby infrastructure. The main advantages are a reduced end-to-end latency, lower backhaul usage through

on-device filtering and compression, improved resilience to intermittent connectivity, and better data locality for privacy-sensitive content. The drawbacks include stringent constraints on the computing power, memory, and energy; thermal throttling risks; increased software maintenance across heterogeneous devices; and the need for model compression or specialization to meet real-time and endurance requirements.

The computational load can be placed entirely on the end device [18], offloaded to proximate fog nodes [15], or split so that a lightweight stage runs onboard and compressed features or cropped imagery are transmitted for downstream processing [3,19]. Each strategy yields a distinct trade-off between onboard inference energy and communication energy at the UAV. Device-centric processing minimizes radio usage, but increases the computing draw; fog-centric designs reduce onboard computing at the cost of continuous or frequent uplink transmission; and hybrid approaches exploit selective, event-driven transmission to balance both.

For computer-vision pipelines using CNNs, quantization [3] and pruning [3,18] are common to reduce the inference energy and memory footprint, typically at the price of some accuracy loss. Using AI-guided pruning, the work by [18] reported energy savings of roughly 40% across several models with accuracy drops below 1%, illustrating that careful compression can preserve the task performance while significantly improving the efficiency on the edge. For vision transformers, adaptive model-partitioning mechanisms and dynamic scaling methods were introduced in [19], leading to reductions in energy consumption and latency.

Beyond model compression, several works explicitly consider the energy balance between communication and computation. The authors of [20] proposed an energy-aware scheduling and offloading algorithm for vehicular fog computing that offloads complex tasks to nearby idle vehicles in latency-sensitive scenarios. Extending this perspective to aerial swarms, a cooperative sensing and transmission scheme for multiple UAVs was designed in [21], jointly optimizing task allocation and transmission power to minimize the total mission time under energy constraints; notably, overlapping sensing tasks can be optimal under certain conditions. Similarly, the authors of [22] jointly designed the UAV trajectory, power allocation, and sensing schedules to reduce the energy required to complete a mission objective. Finally, the adaptive mechanisms for model partitioning proposed in [19,23] reduced the energy consumption, latency, and required computational resources by distributing the workload across a heterogeneous environment.

In contrast to works that minimize the mission time or total energy for a fixed task, our objective was to maximize the mission lifetime in persistent surveillance. We focused on the inference–communication interplay at the single-UAV edge by jointly selecting the YOLO model scale, the detection threshold, and, critically, the inference frequency. This setting is especially relevant for wildfire monitoring in remote areas, where UAVs may operate far from the ground station and the wireless link is both bandwidth- and latency-constrained; transmitting more than necessary can saturate the uplink, increase queueing delays, and degrade the timeliness of actionable alerts. Accordingly, our pipeline follows an event-driven design: onboard inference filters the video stream locally, and only positive detections are transmitted as compact crops with lightweight metadata. This selective transmission scales naturally with the mission duration and fleet size, because the communication load is proportional to the number of detected events rather than to the raw sensor rate, and it also minimizes the workload on the ground by avoiding continuous video review while still preserving human- or AI-based confirmation when needed.

3. Materials and Methods

Section 3.1 describes the pipeline used to derive the FLAME variant employed in this study and the associated training process: the mask conversion to YOLO annotations, multi-frame fire tracking, and fine-tuning of YOLO models. The curated dataset is publicly available in [9]. Section 3.2 then details the energy model used for inference and transmission.

3.1. YOLO Fine-Tuning

3.1.1. Dataset

We constructed a YOLO-format object-detection dataset from FLAME aerial imagery [2,8] by converting pixel-wise fire masks into normalized bounding boxes and by enforcing a temporally aware split. Specifically, we used the ninth FLAME repository (2003 frames at 3480×2160), pairing each frame with its binary fire mask. These are sequential video frames captured by a UAV during a prescribed pile burn in Northern Arizona, USA, rather than independent still images.

For each mask, the connected components were extracted; spatially proximate or overlapping boxes (within 50 px) were merged to avoid over-fragmentation; and very small regions (area < 500 px) were removed to reduce label noise. The labels followed the YOLO convention for a single class (fire), where the coordinates and box dimensions are normalized to $[0, 1]$. Frames and labels were ordered to preserve the chronology, and we produced an 80/20 train/validation split using the earlier portion for training and the later portion for validation to minimize temporal leakage between subsets. For transparency, the dataset released in [9] includes a video overlaying the ground-truth bounding boxes for visual inspection. The video playback frame rate was arbitrarily selected, as the original capture frame rate was not documented and does not affect the underlying annotations.

Because the selected FLAME subset corresponds to a single continuous UAV video sequence acquired during a prescribed pile burn in Northern Arizona, USA, the chronological split does not introduce systematic changes in the geographic location. Visual inspection of the sequence also indicates that the scene, weather, and illumination conditions remain approximately constant throughout the recording, as expected for consecutive frames captured during the same flight segment. Therefore, the chronological split was chosen primarily to reduce temporal leakage between the training and validation data.

3.1.2. Fire Tracking

To be able to evaluate recall at the level of distinct fires rather than per-frame detections, we performed multi-frame tracking and assigned a persistent identifier to each fire across the sequence. Starting from the YOLO-format labels, detections were reconstructed in pixel coordinates and sorted by frame index. Within a sliding temporal window, detections in different frames were linked when at least one spatial-consistency rule held: full inclusion of one box in another, an intersection-over-union (IoU) above 0.2, or a Euclidean center distance below 80 px. A union-find (disjoint-set) structure then merged connected detections into a single track per physical fire (robust to brief disappearances and moderate size/position shifts), and post hoc manual merges corrected rare splits. The output comprises, for each fire, the set of associated detections and its temporal span.

The tracking outcome is illustrated in the dataset release video [9], where each fire ID in the validation set is overlaid and can be visually followed throughout the sequence. The tracking results for the training split are also provided in the release, although they were not used in the experiments reported here and are included mainly to support possible future cross-validation studies. Applied to the validation subset and followed by a quality-control pass, the procedure identified 8 distinct fires. Because the sequence ends abruptly

(a dataset-curation artifact), one fire appears only at the end and is visible in just 13 frames, which is shorter than would be expected in continuous operation because the video ends while the fire is still near the center of the frame.

3.1.3. Training

We fine-tuned the five YOLOv5 scales (n, s, m, l, x) as single-class fire detectors on the training split and evaluated them on the temporally disjoint validation split. Unless otherwise stated, we used default hyperparameters with a 640×640 input size; a batch size of 32; a learning rate of 0.01; and 100 training epochs per model. The default YOLOv5 loss combines a classification term (binary cross-entropy on class logits), an objectness term (binary cross-entropy on candidate boxes), and a localization term (complete IoU). With a single class, the classification head reduces to a one-logit target. Training was executed on a multi-core server equipped with 144 Intel Xeon Gold 5220 processors and 256 GB RAM. All five scales were trained with the same configuration and data to enable a fair comparison of metrics across model sizes.

3.2. Energy Model

We modeled the total energy as the sum of two terms per processed frame: the energy required by deep learning inference on the onboard computer, and the energy to transmit the resulting outputs to ground. The communications term follows a long-term evolution (LTE) user equipment (UE) power model with component-wise contributions proposed by [5]; the inference term was derived from empirical measurements for YOLOv5 models by [6], averaged across two independent measurement strategies. The implementation used in our experiments follows the formulations detailed below.

3.2.1. LTE UE Communications Model

LTE is a widely deployed cellular access technology that provides broadband wireless connectivity with adaptive modulation, power control, and well-characterized energy consumption behavior at the user equipment (UE) level. LTE is commonly used in UAV applications operating beyond short-range radio links, particularly in remote or sparsely populated areas where direct line-of-sight communication to a ground station is not feasible. In such scenarios, the UAV acts as an LTE UE that periodically transmits data to the cellular infrastructure, and the energy cost of this transmission depends on both physical-layer parameters (e.g., transmission power and data rate) and the baseband processing activity. As a result, LTE UE power models provide a suitable abstraction to quantify communication energy consumption in UAV-based sensing pipelines.

Let $m_{Tx}, m_{Rx}, m_{idle} \in \{0, 1\}$ indicate the UE operating mode (uplink, downlink, or idle). According to [5], the UE power is expressed as a mode-weighted sum of the constant, baseband, and radiofrequency (RF) terms:

$$P_{UE} = m_{idle} \cdot P_{idle} + \overline{m_{idle}} \cdot \{P_{co} + m_{Tx} \cdot m_{Rx} \cdot P_{Rx+Tx} + m_{Rx} \cdot [P_{Rx} + P_{RxRF}(S_{Rx}) + P_{RxBB}(R_{Rx}) + m_{2CW} \cdot P_{2CW}] + m_{Tx} \cdot [P_{Tx} + P_{TxRF}(S_{Tx}) + P_{TxBB}(R_{Tx})]\} \quad (1)$$

where S_{Tx} and S_{Rx} are the transmission and receiving power levels (in dBm), P_{idle} represents the power consumption when the UE is in the idle state, and P_{Rx} , P_{Tx} , and P_{Rx+Tx} model the base power that the reception, transmission, and both chains consume when active. R_{Tx} and R_{Rx} are the physical-layer data rates (in Mbit/s), P_{co} captures the board-level and housekeeping consumption, and m_{2CW} activates when two downlink codewords are used, requiring additional power P_{2CW} . In our uplink-only use case (image upload), we set $m_{Tx} = 1$, $m_{Rx} = m_{idle} = 0$, and the transmission energy can therefore be written as

in Equation (2), where T_{Tx} represents the transmission airtime. Idle costs were excluded because they are a fixed consumption that cannot be optimized.

$$E_{UE} = [P_{co} + P_{Tx} + P_{TxRF}(S_{Tx}) + P_{TxBB}(R_{Tx})] \cdot T_{Tx} \quad (2)$$

Following the LTE UE power model parameterization, the constant and mode overheads used were $P_{co} = 1.53$ W and $P_{Tx} = 0.55$ W. Baseband terms were modeled as linear polynomials in the data rate:

$$P_{TxBB}(R_{tx}) = (34.5 + 0.87 R_{tx}) \text{ mW} \quad (3)$$

and RF terms as functions of the relevant power levels:

$$P_{TxRF}(S_{Tx}) = \begin{cases} (-71.3 + 5.50 S_{Tx}) \text{ mW}, & -30 \text{ dBm} \leq S_{Tx} \leq 10 \text{ dBm} \\ (-943 + 117 S_{Tx}) \text{ mW}, & 10 \text{ dBm} < S_{Tx} \leq 23 \text{ dBm} \end{cases} \quad (4)$$

with S_{Tx} clamped to $[-30, 23]$ dBm in accordance with the model's validity range. Equations (3) and (4) correspond to the linear and piecewise-linear fits provided for LTE UE baseband and RF blocks (Tables II–III in the original model [5]) and are the exact forms used in our calculations.

For uplink packets, the energy is power times airtime, modeled as one bounding box crop per transport block. The airtime accounts for protocol headers and forward-error-correction (FEC) overhead, converting an application payload into the actual coded bits sent over the air. We considered 20 bytes of IP (internet protocol) headers, 8 bytes of UDP (user datagram protocol) headers, 2 bytes of PDCP (packet data convergence protocol) headers, 2 bytes of RLC (radio link control) headers, 3 bytes of MAC (medium access control) headers, and a code rate of 1/3. In our experiments, the uplink power was selected to reflect conservative long-range operation (maximum $S_{Tx} = 23$ dBm within model validity) and a bitrate was set to 40 Mbps, as experimentally measured by [24].

The packet sizes were computed from the joint photographic experts group (JPEG) files actually produced by the detector (with the cropped regions being saved using the OpenCV Python library v4.9 with an image quality of 90%), and transformed to over-the-air bits. As will be discussed in the sensitivity analysis, the major conclusions remained unaltered even following substantial changes to the optimization function. This suggests that reasonable changes to these parameters have no critical impact. Figure 1 represents the energy required in mWh (1 mWh = 3.6 J) to transmit each ground-truth bounding box crop from the training set as a function of the bounding box area. The relationship is approximately linear because larger crops contain more pixels and generally produce larger JPEG payloads, increasing the LTE uplink airtime. However, the data are visibly separated into three groups with different slopes. This separation is mainly attributed to differences in the JPEG compression efficiency caused by the image content: a visual inspection of the largest crops, where the separation was most apparent, showed distinct background characteristics for each group. The highest-energy group generally corresponded to close overhead views with visible leaves and branches; the intermediate group to more horizontal forest views where trees are visible, but the fine leaf texture is less prominent; and the lowest-energy group to fire crops over a relatively uniform snow background. Accordingly, the data were fitted with three linear regressions, one for each group. The intercept was fixed to 8.18433×10^{-5} mWh, corresponding to the energy required to transmit an empty packet, including protocol headers. The resulting grouped linear model achieved an overall R^2 of 0.98.

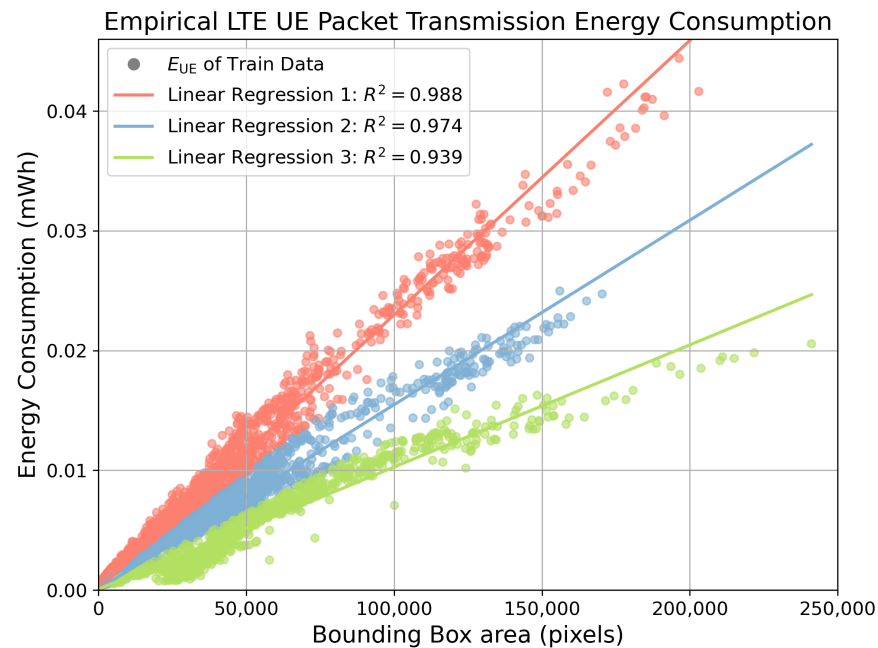


Figure 1. Estimated long-term evolution (LTE) uplink energy required by the user equipment (UE) to transmit cropped detections from the training set as a function of the bounding box area. The data form three approximately linear groups due to differences in the JPEG compression efficiency for different contents. Each group is fitted with a linear regression whose intercept is fixed to the energy required to transmit an empty packet, including protocol headers.

3.2.2. Inference Energy Model

The per-frame inference energy for the five YOLOv5 scales $k \in \{n, s, m, l, x\}$ was derived from published measurements across heterogeneous devices, using the mean of two independent devices reported by [6], NVIDIA Jetson Nano (NJJN) and LynSyn Lite Board (LSLB). For each model k , the per-frame inference energy $E_{\text{inf}}^{(k)}$ is

$$E_{\text{inf}}^{(k)} = \frac{E_{\text{LSLB}}^{(k)} + E_{\text{NJJN}}^{(k)}}{2}, \quad (5)$$

where $E_{\text{LSLB}}^{(k)}$ and $E_{\text{NJJN}}^{(k)}$ are the per-image energy figures from the two devices. The numerical values used were

$$\begin{aligned} [E_{\text{LSLB}}^{(k)}]_{k \in \{n, s, m, l, x\}} &= [0.159, 0.354, 0.804, 1.474, 2.562], \\ [E_{\text{NJJN}}^{(k)}]_{k \in \{n, s, m, l, x\}} &= [0.161, 0.363, 0.822, 1.501, 2.580], \end{aligned} \quad (6)$$

expressed in mWh/frame. These values were used consistently throughout the energy-performance analysis.

3.2.3. Total Energy per Processed Frame

For a given configuration (YOLOv5 scale k , confidence threshold, and inference frequency), the expected energy per processed frame is

$$E_{\text{frame}} = E_{\text{inf}}^{(k)} + \mathbb{E}[E_{\text{UE}}], \quad (7)$$

where the second term $\mathbb{E}[E_{\text{UE}}]$ is the expected uplink energy contribution, conditioned on whether detections occur and on the size and number of crops generated in that frame. Since the inputs are always resized to 640×640 for YOLOv5 inference, $E_{\text{inf}}^{(k)}$ depends on the

chosen model scale, but not on the original image resolution; in contrast, E_{UE} scales with the payload size of the transmitted crops.

The following assumptions were made: (i) The LTE model is applied in uplink-only mode for this application; idle and downlink reception terms are omitted from the optimization since no downlink traffic is generated and the idle state energy cannot be optimized. (ii) The packetization overhead and FEC are explicitly accounted for, as reported in Section 3.2.1. (iii) Inference energies are single-image measurements averaged across two models to mitigate measurement bias; they were used as model-dependent variables in our simulations. (iv) Communications parameters (transmission power, physical-layer rate) can be tuned to scenario constraints; the results reported here used conservative values consistent with long-range operation.

4. Results

This section reports the performance of the five YOLOv5 scales trained on our FLAME-derived dataset and quantifies the detection–energy trade-offs that motivated the final configuration. We first summarize the training outcomes and model-level metrics, emphasizing recall as the operationally critical objective for early fire detection. We then analyzed frame-level behavior as a function of the confidence threshold and model size, extended the analysis to temporally aware detection under a reduced inference frequency, and finally selected the optimal operating point using a unified objective that jointly considers event-level detection probabilities and energy consumption. The main parameters used in this section are described in Table 1.

Table 1. Key parameters used by the optimization problem.

Symbol	Definition
k	YOLOv5 model scale (n/s/m/l/x).
τ	Confidence threshold used to accept detections.
s	Inference stride: one inference every s frames.
i	Index of a tracked fire (event).
A_i	Annotation count linked to fire i (effective detection opportunities).
p	Per-opportunity detection probability for a given configuration (k, τ); modeled as common across fires under that configuration for the theoretical approach.
n_i	Expected number of processed opportunities for fire i at stride s (approx. $n_i \approx A_i/s$).
P_i	Probability of detecting fire i at least once (event-level) in the validation set.
P_{all}	Probability of detecting all tracked fires in the validation set.
P_{almost}	Probability of detecting at least all but one tracked fire in the validation set.
$E_{\text{inf}}^{(k)}$	Inference energy per processed frame using model k expressed in mWh/frame.
E_{UE}	Transmission energy per processed frame (depends on detections produced) expressed in mWh/frame.
E_{frame}	Energy per processed frame expressed in mWh/frame.
a, b, c	Objective weights for P_{all} , P_{almost} , and energy, respectively.
J	Objective function.

4.1. Training Outcomes Across YOLOv5 Scales

All five YOLOv5 variants (n, s, m, l, x) were trained for 100 epochs on the same training split and evaluated on the temporally disjoint validation split. To mitigate overfitting to the last epoch, the checkpoint with the highest validation mAP was selected for each model.

Table 2 summarizes the training time, the parameter count, the epoch at which the best checkpoint was reached, and the corresponding recall on the validation split (measured with the 50% inclusion rule defined in Equation (8) below).

Table 2. You only look once (YOLO) v5 training summary on the FLAME-derived dataset. The recall confidence interval (CI) was calculated using the Wilson method for a binomial distribution.

Model	Training Time	Parameters	Best Epoch	Recall	95% CI
YOLOv5n	3.06 h	1.77 M	92	0.823	[0.805, 0.839]
YOLOv5s	6.37 h	7.02 M	77	0.844	[0.827, 0.860]
YOLOv5m	12.68 h	20.87 M	41	0.863	[0.847, 0.877]
YOLOv5l	25.30 h	46.13 M	91	0.842	[0.825, 0.858]
YOLOv5x	39.64 h	86.21 M	92	0.845	[0.828, 0.861]

As expected, larger models require longer training and, beyond YOLOv5m, yielded diminishing or even negative returns in recall on this dataset—likely reflecting a limited data scale relative to the model capacity. A fire is counted as detected if at least 50% of its ground-truth bounding box is covered by a predicted box in the frame; recall and precision follow the standard definitions based on true positives (TPs), false positives (FPs), and false negatives (FNs):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (8)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (9)$$

Note that recall cannot be used directly as a training loss (a trivial always-positive classifier would maximize Equation (8)), hence the use of the default YOLOv5 multi-term loss and mAP-based checkpoint selection. Training graphs for YOLOv5n are provided in Figures 2–4.

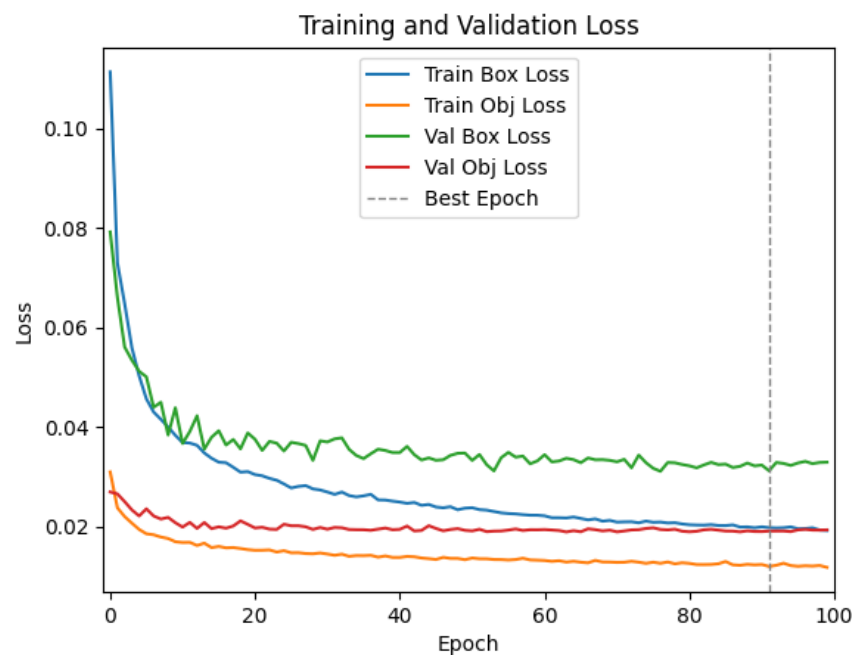


Figure 2. Training and validation loss per epoch during YOLOv5n training.

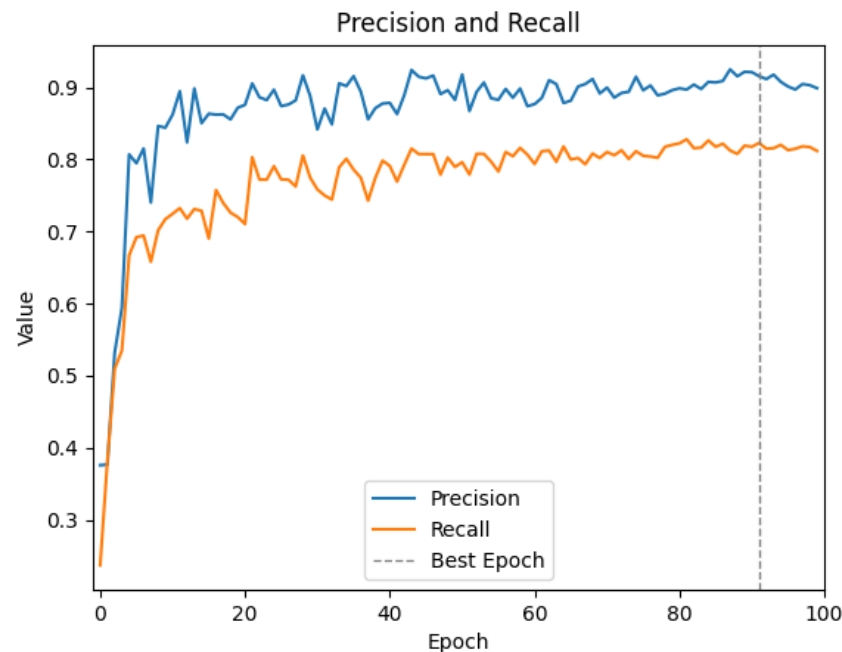


Figure 3. Precision and recall on the validation subset per epoch during YOLOv5n training.

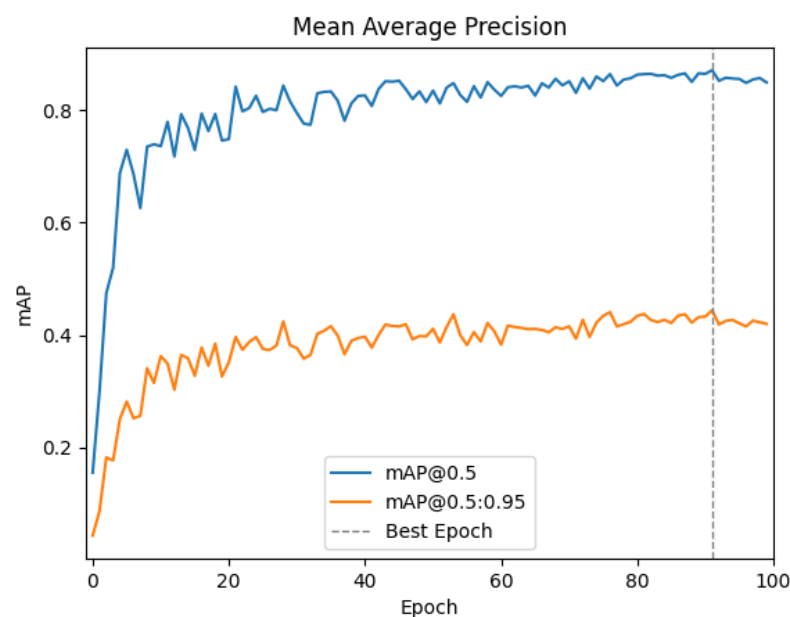


Figure 4. Mean average precision per epoch during YOLOv5n training.

4.2. Frame-Level Detection and Energy Trade-Offs

To study the detection–energy balance at the frame level, we evaluated combinations of the model scale and the confidence threshold on the validation set. Lower thresholds prioritized recall, reflecting the asymmetric costs of missed detections versus false positives; we analyzed the threshold $\tau \in \{0.02, 0.03, 0.05, 0.07, 0.10, 0.20, 0.30, 0.45\}$, which encompasses a comprehensive array of reasonable values. For each pair (k, τ) , we report the recall under the 50% inclusion rule in Figure 5 and the total energy per processed frame, as defined in Section 3.2, in Figure 6. The results were calculated by averaging over the 401 frames that make up the validation dataset.

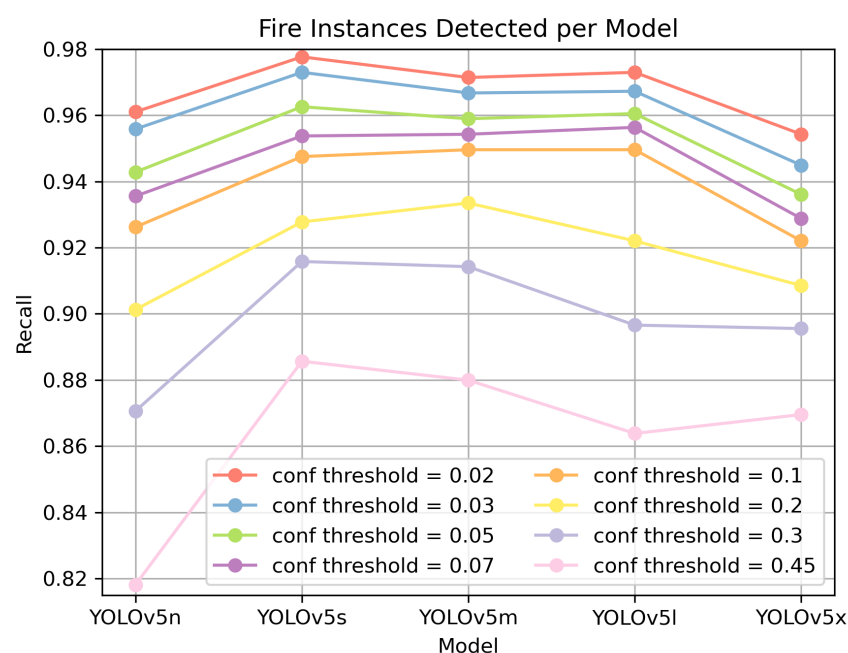


Figure 5. Frame-level recall across model scales different confidence thresholds.

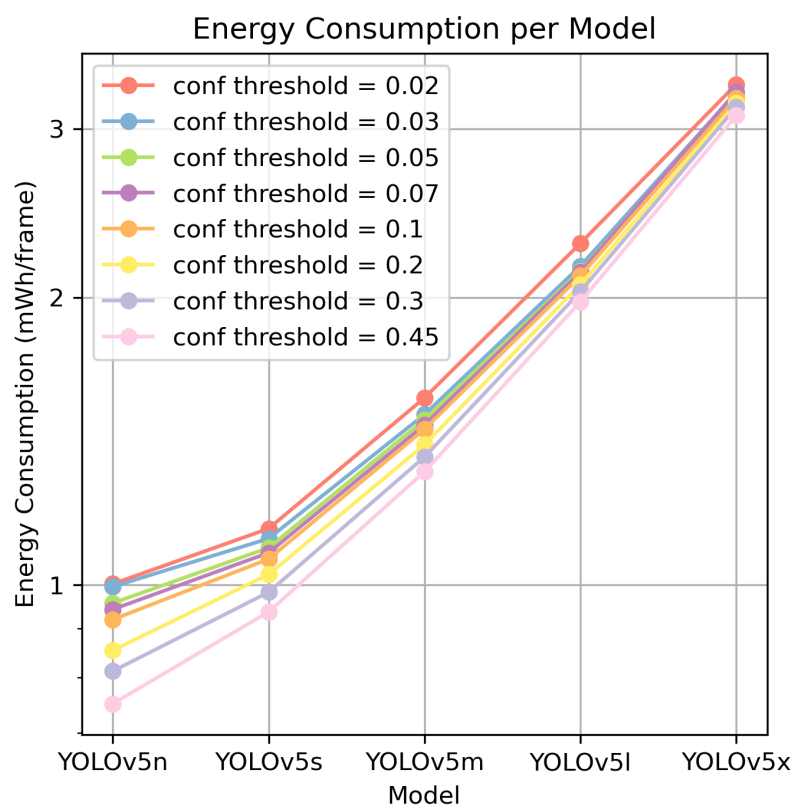


Figure 6. Total energy per frame ($E_{\text{frame}} = E_{\text{inf}} + E_{\text{UE}}$) across confidence thresholds for each scale.

Three consistent trends emerged. First, moving from YOLOv5n to YOLOv5s yielded the only recall gain that remained robust across thresholds. Scaling further to m/l/x did not produce consistent improvements under our validation criterion and dataset size; at most thresholds, recall even degraded for the largest model. This behavior is likely related to the limited dataset size, which increased the risk of overfitting, a common limitation in UAV-based fire detection. Second, lowering the confidence threshold increased E_{UE}

by producing more positive detections (and therefore, more crops to transmit), whereas E_{inf} depended only on the chosen model. Consequently, E_{frame} rose at a low τ because of communication costs and rose with model size because of inference costs, with the model-size effect dominating the smaller variations due to communication overhead. Third, the inference time grows with the model size, ranging from 0.34 s for YOLOv5n to 0.56 s for YOLOv5x on the previously described hardware.

In summary, the per-frame analysis indicates that a compact model with a relatively low threshold achieved a high recall at a substantially lower energy cost than larger variants. This motivates focusing on compact detectors and seeking additional efficiency gains through temporal strategies (e.g., reducing inference frequency with tracking-aware evaluations), which we explore in the following subsection.

4.3. Temporally Aware Detection

In real deployments, a given fire typically spans multiple consecutive frames as the UAV moves and the camera continuously captures the scene. This temporal redundancy can be exploited to reduce the inference frequency without substantially degrading the event-level recall: it is sufficient to detect the fire in any one of the frames in which it appears to confirm its presence. Table 3 summarizes the eight tracked fires identified in the validation split, including their frame ranges, their durations (in frames), and the total number of linked annotations assigned to each track following the process described in Section 3.1.2. The video ends abruptly due to dataset curation; consequently, the last fire appears only at the tail of the sequence and is visible for 74 frames, with only 13 linked annotations.

Table 3. Tracking results for the fires that appear in the validation subset. The duration is measured in frames (inclusive of endpoints), while the annotation count denotes the number of linked mask-derived boxes assigned to each track.

i	Annotation Count A_i	Frame Range	Duration
24	288	1432–1719	288
25	508	1449–1947	499
26	631	1460–1951	492
27	396	1564–1959	396
28	234	1652–1978	327
29	308	1668–2002	335
30	69	1668–1839	172
31	13	1927–2000	74

We evaluated inference strides $s \in \{1, 2, 3, 4, 5, 6\}$, where $s = 1$ denotes inference on every frame and $s > 1$ denotes processing one out of every s frames. A fire was counted as detected if at least one processed frame contained a valid detection. To obtain generalizable estimates across strides, we introduced a simple theoretical estimator in which detection opportunities are modeled as independent Bernoulli trials with a probability p , and the phase between the sampling grid and the video start is assumed to be random. Here, p depends on the chosen detector (model scale) and confidence threshold, but is assumed to be the same for all fires under a fixed configuration. In practice, p is estimated from the validation set using the same matching rule used throughout this paper: a fire is detected if at least 50% of its ground-truth box is included in a predicted box by the YOLO model.

Because occlusions, camera rotation, and mask fragmentation can cause a single physical fire to yield multiple linked annotations within the same frame (e.g., disjoint connected components producing multiple boxes), the annotation count A_i can exceed the frame duration for some tracks. Importantly, this phenomenon also occurs during training and is therefore reflected in the empirical estimation of p : multiple same-frame annotations

represent multiple detection opportunities that the detector is exposed to under the same data characteristics. We thus used A_i as the number of effective detection opportunities associated with fire i . Under stride s , the expected number of processed opportunities was approximated as $n_i \approx A_i/s$. Under this heuristic, the probability P_i of the fire i being detected in the validation set is

$$P_i = 1 - (1 - p)^{A_i/s}, \quad \forall i \in \{24, \dots, 31\}. \quad (10)$$

The approximation $n_i \approx A_i/s$ is exact in two limiting cases that motivate its use as a practical estimator: (i) If the fire appears continuously with exactly one annotation per frame, then A_i equals the duration and sampling every s frames yields $\lfloor A_i/s \rfloor \leq n_i \leq \lceil A_i/s \rceil$ for any offset, and $\mathbb{E}[n_i] = A_i/s$ when averaging over offsets. (ii) If the A_i opportunities are distributed uniformly at random over frames, then each opportunity is sampled with probability $1/s$, so $\mathbb{E}[n_i] = A_i/s$ with $0 \leq n_i \leq A_i$. In real scenarios, the distribution of n_i is unknown, as it depends on many different factors and is difficult to estimate beforehand. Therefore, we take $n_i = A_i/s$ as a simple, but realistic, approximation. In any case, this estimator should be interpreted as a heuristic analytical approximation rather than as an exact probabilistic model of the UAV video sequence. Consecutive frames are temporally correlated, and therefore, the detection outcomes in nearby frames are not independent. The estimator is nevertheless useful as a simple upper-bound approximation for analyzing how the number of available detection opportunities changes with the inference stride. For this reason, it is complemented by the empirical offset sweep, which directly evaluates all possible sampling phases on the actual video sequence and does not rely on the independence assumption.

To quantify the effect of the independence assumption, we compared the theoretical estimator with the empirical offset sweep across the evaluated configurations. Since consecutive UAV video frames are temporally correlated, the independent-Bernoulli model is expected to overestimate the benefit of repeated detection opportunities. This behavior was observed in the present dataset: when comparing the expected fraction of detected fire events, the theoretical estimator was, on average, 0.05 ± 0.05 higher than the empirical offset-sweep result, with differences ranging from 0 to 0.12. The discrepancy was mainly associated with the shortest and most distant fire track, fire 31, which provides fewer reliable detection opportunities; the remaining tracked fires were detected in all empirically tested scenarios. Consequently, the empirical value of P_{almost} was equal to 1 for all considered configurations, while differences in P_{all} were determined exclusively by whether this shortest fire was detected for the different sampling offsets. Therefore, the theoretical estimator was retained only as an interpretable analytical upper-bound approximation for this dataset, whereas the empirical offset sweep was used as the main evidence for final configuration selection.

When inference is applied to every frame ($s = 1$), the probability of detecting each of the eight fires is virtually one across all model scales and thresholds considered. As the stride increases, event-level recall gradually degrades—more noticeably at higher confidence thresholds—because short-lived fires offer fewer processed opportunities. This event-level metric is higher than the per-frame recall for persistent fires when $n_i > 1$, because the event can still be detected after multiple attempts even when $s > 1$. Figure 7 shows the estimated probability of detecting every fire with $s = 6$ for all considered models and confidence thresholds. If we allow up to one fire to remain undetected, Figure 8 suggests that the stride can be increased to as much as $s = 35$ while preserving a similar outcome under the theoretical estimator.

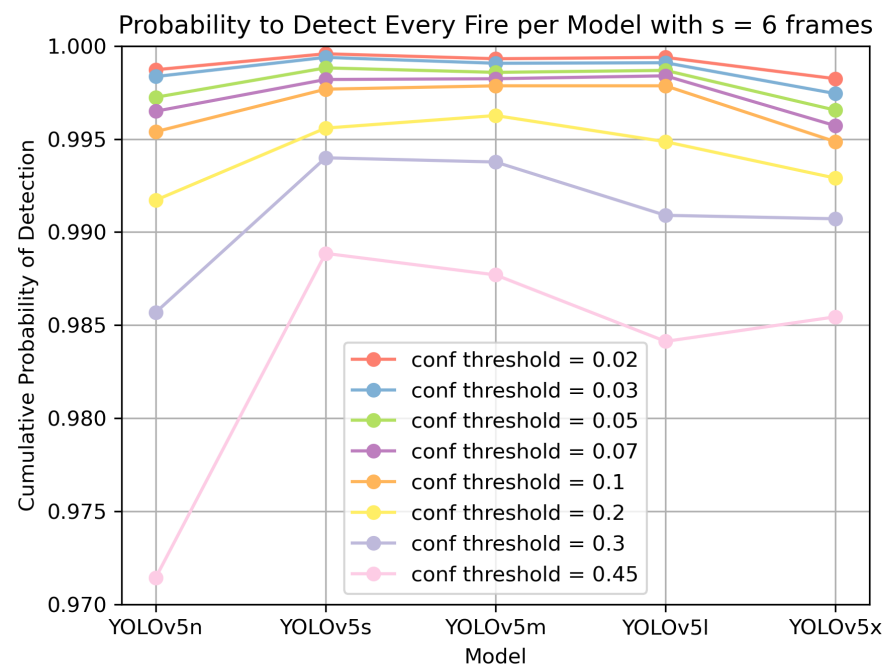


Figure 7. Expected detection probability for inference stride $s = 6$ frames at different thresholds.

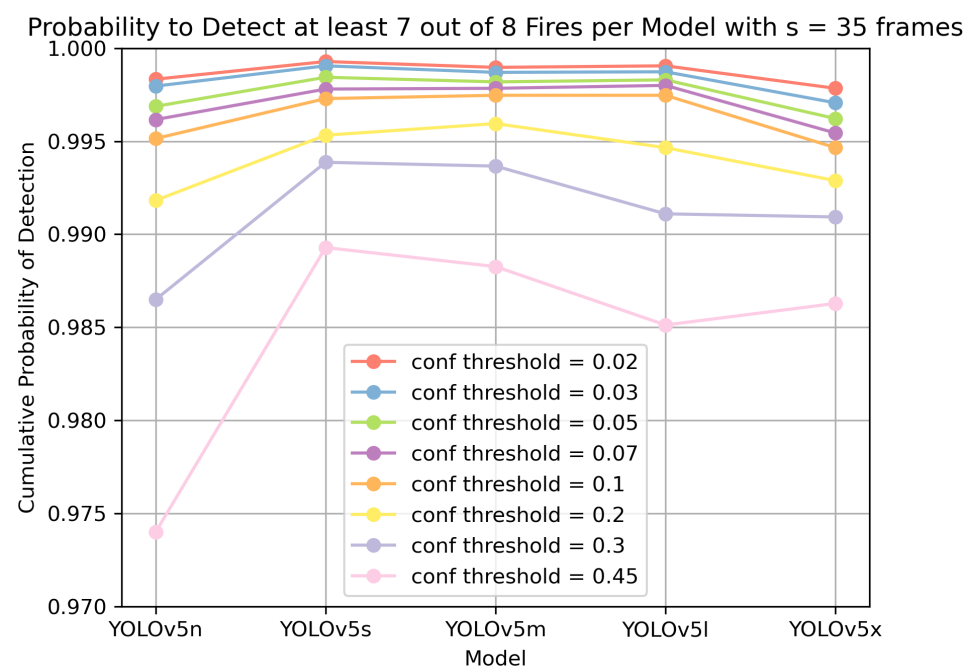


Figure 8. Expected probability of detecting at least all but one fire for inference stride $s = 35$ frames at different thresholds.

To complement the theoretical estimator, Figure 9 reports the number of fires actually detected for $s = 5$, averaged across all six possible offsets, for every model and confidence threshold considered. Specifically, every fire $i \in \{24, \dots, 30\}$ was detected in all scenarios. The only fire that occasionally failed to be detected was fire 31, which is the shortest track in the validation split and would likely have appeared for longer if the video had not ended abruptly.

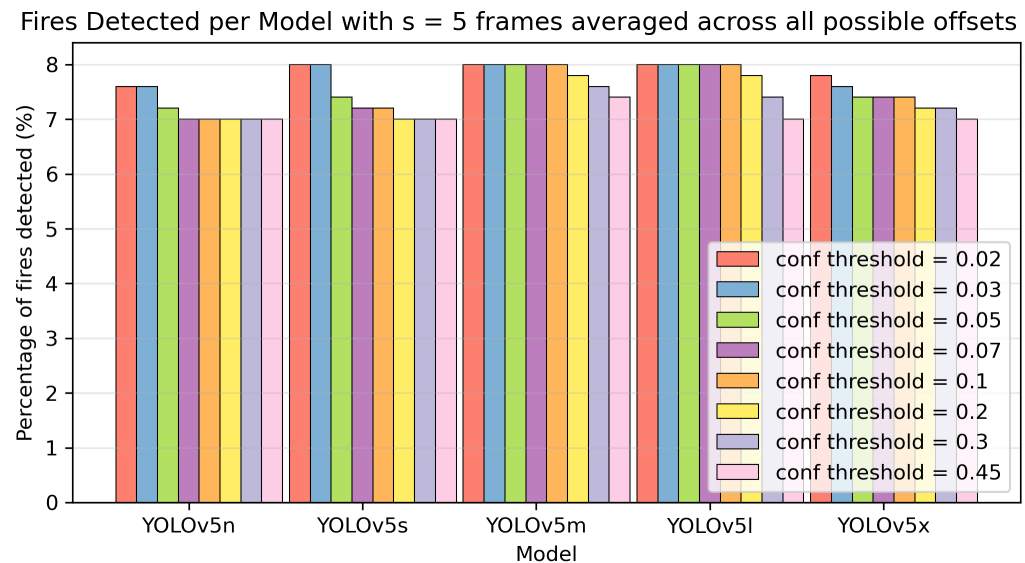


Figure 9. Average number of fires detected for inference stride $s = 5$ frames at different thresholds, averaged across all possible offsets.

Reducing the inference frequency yielded a near-linear reduction in total energy: with stride s , the average inference term scaled as E_{inf}/s , and the communication term also decreased roughly proportionally because positive detections (and thus, transmitted crops) occurred in fewer processed frames. Temporal subsampling is therefore an effective, low-complexity lever for extending mission endurance: choosing $s > 1$ preserves a near-perfect event-level recall for most fires in this dataset, with the primary vulnerability confined to very short tracks at the end of the sequence. Combined with a conservative confidence threshold and a compact detector, this strategy underpins the energy-recall operating points analyzed in Section 4.4.

4.4. Energy-Aware Parameter Selection

To select an operating point that balances detection and energy, we combined three metrics: (i) the probability of detecting all fires, P_{all} ; (ii) the probability of detecting at least all but one fire, P_{almost} ; and (iii) the expected energy E_{frame} (mWh per frame). We evaluated each configuration (model k , threshold τ , stride s) using two complementary estimators:

- **Theoretical:** For each tracked fire i and per-frame detection probability p under the chosen configuration, the event-level detection probability is approximated as described in Section 4.3. In this estimator, A_i denotes the annotation-count proxy for the number of detection opportunities associated with fire i . From P_i , we obtain P_{all} and P_{almost} over the eight fires, computed from the distribution of the number of detected fires under the independence and random-phase assumptions. These event-level probabilities should therefore be interpreted as a heuristic approximation, because consecutive UAV frames are temporally correlated.
- **Empirical:** For each stride s , we sweep all possible offsets $o \in \{0, \dots, s-1\}$, run inference on frames $\{t : t \equiv o \pmod{s}\}$, and measure the percentage of offsets for which (a) all fires are detected (P_{all}) or (b) at most, one fire is missed (P_{almost}). This directly reflects the configuration's robustness to the sampling phase without relying on independence assumptions.

With these ingredients, we defined the objective J that we sought to maximize. Because very low detection probabilities are unacceptable in practice, we adopted a concave utility

that is deliberately more sensitive to small probabilities, and the optimization function J is defined in Equation (11).

$$J = a \log P_{\text{all}} + b \log P_{\text{almost}} - \frac{c}{200} E_{\text{frame}} \quad (11)$$

Taking logarithms strongly penalizes low detection probabilities. The energy term is normalized by dividing it by 200 to approximately match the standard deviation of $\log P_{\text{all}}$, preventing the energy term from dominating the objective solely because of its units and numerical range. To assess the robustness, we ran a sensitivity analysis over the weights by maximizing Equation (11) on a grid $a, b, c \in \{1, 2, \dots, 10\}$ (10^3 combinations). For the theoretical estimator, a compact model (YOLOv5n or YOLOv5s) was always the selected model, $s \in \{4, 6\}$ the selected stride, and the smaller confidence threshold $\tau = 0.02$ was the best option on 98% of the analyzed scenarios. Table 4 reports how often each configuration maximized J . The empirical estimator produced a similar conclusion, with YOLOv5s always emerging as the best choice at a stride of $s = 5$ and a confidence threshold of 0.03.

Table 4. Configuration that maximizes the objective under weight sweeps ($a, b, c \in [1, 10]$) for the theoretical estimator. Percentages indicate the fraction of weight settings for which the configuration is optimal.

% of Weight Settings	Model	Threshold	Inference Stride
53.0%	YOLOv5n	0.02	$s = 4$
37.0%	YOLOv5s	0.02	$s = 6$
5.0%	YOLOv5s	0.02	$s = 4$
3.0%	YOLOv5n	0.02	$s = 6$
2.0%	YOLOv5s	0.03	$s = 4$

These results indicate that a compact model with a reduced inference frequency dominates across a broad range of preferences, whereas the optimal confidence threshold still varies with the relative emphasis on recall versus energy. In other words, the weight sweep shows that the preferred model size and stride are robust across a broad range of values, while the threshold remains the most sensitive parameter. The value of the optimization function J as a function of τ and s is represented in Figure 10.

To quantify the practical benefit of the proposed optimization, we compared the expected energy consumed per frame under three operating modes: (i) the configuration selected by the empirical optimization procedure (YOLOv5s with $\tau = 0.03$ and an inference stride $s = 5$), (ii) a high-detection baseline chosen to maximize the probability of detecting all fires, and (iii) the continuous transmission of full images without onboard inference. In our experiments, many parameter combinations achieved $P_{\text{all}} = 1$ during the empirical offset sweep (using double-precision floating-point arithmetic). We therefore selected as the high-detection baseline YOLOv5n with $\tau = 0.03$ and continuous inference ($s = 1$), since it minimizes energy consumption within the set of $P_{\text{all}} = 1$ candidates. For the no-inference case, the reported transmission energy treats frames as individual JPEG images and does not apply video coding, nor does it consider additional headers due to packet segmentation. As shown in Table 5, the optimized configuration reduced the expected energy per frame by a factor of 4.4 relative to the high-detection baseline when running onboard inference, at the cost of a 0.03% reduction in P_{all} . Even if not accounting for frame subsampling, the optimized approach leads to a 33% reduction in energy compared to full-frame transmission.

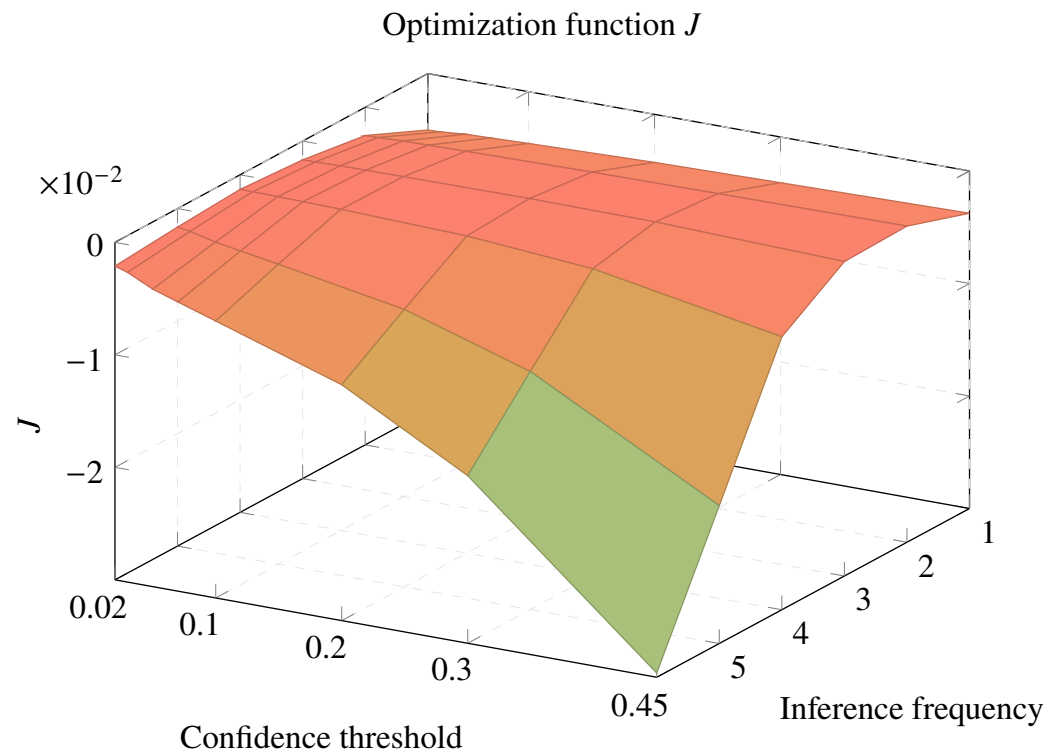


Figure 10. Optimization function J for YOLOv5n with weights $a = b = c = 1$.

Table 5. Energy–detection trade-off between an empirically optimized configuration and a high-detection baseline, with continuous full-frame transmission shown for reference. The energy per processed frame is the sum of the energy for one inference and the mean transmission energy associated with the detections produced in a processed test frame; the energy per frame divides this value by the inference stride s . Both inference-based configurations detected all fires in the validation set.

Configuration	Energy per Processed Frame (mWh/Frame)	Energy per Frame (mWh/Frame)	P_{all} (%)
Empirically optimized J	$0.358 + 0.760$	0.224	99.97
High-detection baseline	$0.160 + 0.836$	0.996	100.00
Full-frame transmission	$0 + 1.673$	1.673	—

Finally, it is important to consider that the optimization performed in this work targeted the onboard visual detection and communication link, namely YOLO-based inference and the uplink transmission of visual evidence. It does not explicitly model propulsion, gimbal actuation, camera sampling, avionics, or other baseline UAV subsystems. Nevertheless, reducing the visual-link energy can still have a relevant effect on the overall power budget. The FLAME dataset was acquired using a DJI Phantom 3 Professional [2], with a reported battery capacity of 68 Wh [25] and an effective flight time of approximately 18 min [2], which corresponds to an average platform power of about 227 W. For comparison, the estimated power required by the visual link decreased from approximately 175 W and 361 W in the full-frame transmission scenario at 29 FPS and 60 FPS, respectively, corresponding to the frame-rate limits reported by Shamsoshoara et al. [2], to approximately 23 W and 48 W with the configuration selected by the objective function. Under this illustrative power-budget calculation, adding the visual-link power to the estimated baseline platform power results in a total power reduction of approximately 38% at 29 FPS and 53% at 60 FPS. These values should be interpreted as scenario-dependent estimates rather than direct flight-endurance measurements, since the exact endurance improvement depends

on the airframe, payload, camera frame rate, onboard computing hardware, wireless link, and mission profile.

4.5. Deployment Configuration

To operationalize the pipeline described in this work, the UAV should run onboard YOLO inference only at a controlled cadence and transmit data only when necessary. Based on the optimization results, we recommend configuring the edge detector as follows. (1) Select a compact model (YOLOv5n/s) to minimize the inference energy while preserving high event-level detection. (2) Use a low confidence threshold (e.g., $\tau \in [0.02, 0.03]$) to bias the system toward higher recall, accepting that occasional false alarms primarily increase the communication cost rather than increasing safety risk. (3) Apply temporally subsampled inference with stride s (typically $s \in \{4, 5, 6\}$ on the selected dataset) so that only one out of every s frames is processed; this reduces both the inference and transmission energy proportionally to $1/s$ while maintaining near-unity event-level detection for fires with a moderate temporal extent. (4) When a detection occurs, transmit only compact evidence to the ground station: cropped regions of interest around detected fires plus lightweight metadata (timestamp, UAV pose/GPS, and optionally confidence score), and do not transmit negative frames. This event-driven design keeps uplink usage bounded and reduces the latency in remote-area operation, where the bandwidth degrades as the UAV moves farther from the base.

5. Discussion

This work presented an energy-aware framework for onboard wildfire detection with UAVs that jointly model and optimize deep learning inference and wireless transmission. We (i) derived a YOLO-ready variant of the FLAME dataset with bounding box annotations and fire tracking to enable event-level evaluation; (ii) fine-tuned five YOLOv5 scales on this dataset; (iii) modeled energy using literature-based inference costs and an LTE UE power model for communications; and (iv) searched over the model scale, confidence threshold, and inference frequency to identify operating points that balance the detection performance and energy consumption.

The main findings are as follows. First, compact detectors dominate the detection–energy trade-off on this dataset: moving from YOLOv5n to YOLOv5s yields only a modest recall gain, while larger models (m/l/x) offer no consistent improvement and substantially increase the inference energy. Second, communication energy rises as the confidence threshold decreases, owing to more positive detections and transmissions, although in our analysis, this rise remained secondary to the inference penalty incurred by larger models. Third, temporal redundancy can be exploited: reducing the inference frequency (processing one every s frames) lowers both the inference and communication energy approximately in proportion to $1/s$, while preserving high event-level recall. Using both a heuristic theoretical estimator and an empirical offset sweep, the preferred operating region consistently lies at small model sizes and low thresholds. Under the unified objective that combines the probabilities of detecting all fires or all but one fire with the expected energy, the recommended configuration is a compact YOLO network with a low confidence threshold and a moderate stride; the exact threshold choice depends on the relative weighting of recall versus energy.

While $s \in \{4, 5, 6\}$ is typically optimal for endurance on the selected dataset, the stride should be adjusted to platform dynamics: a higher UAV speed or a faster yaw reduces the number of frames in which a fire is visible, which leads to fewer processed frames and therefore favors a smaller stride to preserve P_{all} . Conversely, during slow loiter or orbit, a larger stride can extend flight time with negligible impact on event-level recall. In all

cases, transmitting bounding box crops with metadata provides visual confirmation at the ground station while keeping the communication energy bounded.

The sensitivity analysis shows that the optimization outcome is relatively stable across a broad range of weight settings. Consequently, further optimization with standard compression techniques is more likely to improve absolute endurance than to overturn the qualitative operating-point trend observed here. For example, pruning methods such as the one proposed in [18] can markedly reduce the inference energy with negligible accuracy loss, but given the dominance of compact models in our setting, such gains would mainly improve the absolute endurance rather than fundamentally shift the preferred operating region.

Some limitations remain. Our empirical analysis was based on a single-flight dataset of prescribed burns and literature-derived energy models; although the LTE and inference models capture first-order behavior, in situ measurements on the target airframe and radio would refine absolute energy estimates. The use of a single UAV sequence also limits the generalization of the numerical results to other locations, vegetation types, fire scales, lighting conditions, weather conditions, flight altitudes, camera viewpoints, sensing modalities, and UAV platforms. Therefore, the specific optimal configuration reported in this work should be recalibrated when different datasets, sensors, onboard hardware, or communication links are used. Nevertheless, the broader conclusions are expected to remain useful across scenarios. As future work, the proposed framework should be evaluated on additional RGB and infrared aerial fire datasets and cross-dataset validation should be performed when temporally annotated UAV sequences become available.

In addition, the theoretical event-level estimator relies on simplifying assumptions about independence and effective detection opportunities, so the empirical offset sweep should be regarded as the more reliable indicator. As future work, the generality can be further tested by extending the evaluation to additional sensing modalities (e.g., IR), more diverse terrains and fire scales, and alternative links, such as 5G (fifth-generation mobile technology) or sub-gigahertz systems. On the algorithmic side, jointly adaptive policies (dynamically adjusting the threshold and stride based on motion cues, scene dynamics, or recent detections) are a promising direction, as are lightweight onboard tracking and intelligent frame scheduling (e.g., repeating inference in a nearby frame when a score lies near the threshold). Multi-UAV cooperation could leverage overlapping views to safely increase the stride while sustaining high event-level recall. Finally, testing the proposed energy-aware detection framework onboard real UAVs during operational fire-detection missions or controlled prescribed-burn campaigns would further assess its robustness under realistic flight dynamics, communication conditions, and environmental variability.

Author Contributions: Conceptualization, R.A.-C.; methodology, F.-J.A.-A.; software, F.-J.A.-A.; validation, A.-J.G.-S.; formal analysis, J.G.-H.; investigation, R.A.-C.; writing—original draft preparation, F.-J.A.-A.; writing—review and editing, R.A.-C. and J.G.-H.; visualization, J.G.-H.; supervision, J.G.-H. and A.-J.G.-S.; project administration, A.-J.G.-S.; funding acquisition, R.A.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the grant PID2023-148214OB-C21, funded by MICIU/AEI/10.13039/501100011033, and by FEDER/EU. This work was also supported in part by the grant PDC2025-165710-C22 funded by MCIN/AEI/10.13039/501100011033. This work was also supported by the grant PCI2024-153485 funded by MICIU/AEI/10.13039/501100011033 and by the European Union. This research was also funded by the PRIMA Programme under grant agreement No. 2431 (FUSION: Comprehensive and sustainable solution to minimize food loss and waste and promote food security in the Mediterranean region). The work of Francisco-Jose Alvarado-Alcon was supported by the Spanish Ministry of Universities under grant FPU22/00316.

Data Availability Statement: The data used in the study are openly available in Zenodo at <http://hdl.handle.net/10317/21516> [9] (published on May 2026), which extends the previous work by Shamsoshoara et al. [8] (accessed on 15 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
IR	Infrared
JPEG	Joint Photographic Experts Group
LSLB	LynSyn Lite Board
LTE	Long Term Evolution
NJN	NVIDIA Jetson Nano
RF	Radiofrequency
RNN	Recurrent Neural Network
UAV	Unmanned Aerial Vehicle
UE	User Equipment
YOLO	You Only Look Once

References

1. Bouguettaya, A.; Zarzour, H.; Taberkit, A.M.; Kechida, A. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. *Signal Process.* **2022**, *190*, 108309. [\[CrossRef\]](#)
2. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [\[CrossRef\]](#)
3. Saqib, E.; Leal, I.S.; Shallari, I.; Jantsch, A.; Krug, S.; O’Nils, M. Optimizing the IoT Performance: A Case Study on Pruning a Distributed CNN. In Proceedings of the 2023 IEEE Sensors Applications Symposium (SAS), Ottawa, ON, Canada, 18–20 July 2023; pp. 1–6. [\[CrossRef\]](#)
4. Casas, E.; Ramos, L.; Bendek, E.; Rivas-Echeverria, F. YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios. *J. Image Graph.* **2024**, *12*, 127–136. [\[CrossRef\]](#)
5. Jensen, A.R.; Lauridsen, M.; Mogensen, P.; Sørensen, T.B.; Jensen, P. LTE UE Power Consumption Model: For System Level Energy and Performance Optimization. In Proceedings of the 2012 IEEE Vehicular Technology Conference (VTC Fall), Quebec City, QC, Canada, 3–6 September 2012; pp. 1–5. [\[CrossRef\]](#)
6. Matic, I.; de Lemos, F.; Machado, P.; Ihianle, I.K.; Adama, D.A.; Bird, J.J. Estimating the Power Consumption of Heterogeneous Devices when performing AI Inference. *arXiv* **2023**, arXiv:2207.06150. [\[CrossRef\]](#)
7. Jocher, G. YOLOv5 by Ultralytics. 2020. Available online: <https://zenodo.org/records/7347926> (accessed on 15 June 2026).
8. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.; Blasch, E. The FLAME dataset: Aerial Imagery Pile burn detection using drones (UAVs). 2020. Available online: <https://iee-dataport.org/open-access/flare-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs> (accessed on 15 June 2026).
9. Alvarado-Alcon, F.J.; Asorey-Cacheda, R.; Garcia-Haro, J.; Garcia-Sanchez, A.J. Replication data for “Optimizing Energy and Event-Level Recall in UAV Fire Detection via YOLOv5 and Selective Transmission”. 2026. Available online: <https://repositorio.upct.es/entities/publication/89e48200-7b6d-4526-bf9a-e4c438489aee> (accessed on 15 June 2026).
10. Chen, X.; Hopkins, B.; Wang, H.; O’Neill, L.; Afghah, F.; Razi, A.; Fulé, P.; Coen, J.; Rowell, E.; Watts, A. Wildland Fire Detection and Monitoring Using a Drone-Collected RGB/IR Image Dataset. *IEEE Access* **2022**, *10*, 121301–121317. [\[CrossRef\]](#)
11. Mashraqi, A.M.; Asiri, Y.; Algarni, A.D.; Abu-Zinadah, H. Drone imagery forest fire detection and classification using modified deep learning model. *Therm. Sci.* **2022**, *26*, 411–423. [\[CrossRef\]](#)
12. Luan, T.; Zhou, S.; Liu, L.; Pan, W. Tiny-Object Detection Based on Optimized YOLO-CSQ for Accurate Drone Detection in Wildfire Scenarios. *Drones* **2024**, *8*, 454. [\[CrossRef\]](#)
13. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [\[CrossRef\]](#)

14. Goyal, S.; Shagill, M.; Kaur, A.; Vohra, H.; Singh, A. A yolo based technique for early forest fire detection. *Int. J. Innov. Technol. Explor. Eng.* **2020**, *9*, 1357–1362. [\[CrossRef\]](#)
15. Yandouzi, M.; Berrahal, M.; Grari, M.; Boukabous, M.; Moussaoui, O.; Azizi, M.; Ghoumid, K.; Elmiad, A.K. Semantic segmentation and thermal imaging for forest fires detection and monitoring by drones. *Bull. Electr. Eng. Inform.* **2024**, *13*, 2784–2796. [\[CrossRef\]](#)
16. Sungheetha, A.; Sharma, R. Real time monitoring and fire detection using internet of things and cloud based drones. *J. Soft Comput. Paradig.* **2020**, *2*, 168–174. [\[CrossRef\]](#)
17. Uddin, M.N.; Sakibul Islam Sakib, M.; Nawer, S.; Mohona, R.T. Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety. In Proceedings of the 2023 26th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 13–15 December 2023; pp. 1–6. [\[CrossRef\]](#)
18. Zawish, M.; Ashraf, N.; Ansari, R.I.; Davy, S. Energy-Aware AI-Driven Framework for Edge-Computing-Based IoT Applications. *IEEE Internet Things J.* **2023**, *10*, 5013–5023. [\[CrossRef\]](#)
19. Katare, D.; Zhou, M.; Chen, Y.; Janssen, M.; Ding, A.Y. Energy-Aware Vision Model Partitioning for Edge AI. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing; SAC'25*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 671–678. [\[CrossRef\]](#)
20. Liang, C.; Zhao, Y.; Gao, Z.; Cheng, K.; Wang, B.; Huang, L. EALSO: Joint energy-aware and latency-sensitive task offloading for artificial Intelligence of Things in vehicular fog computing. *Wirel. Netw.* **2025**, *31*, 583–599. [\[CrossRef\]](#)
21. Meng, K.; He, X.; Wu, Q.; Li, D. Multi-UAV Collaborative Sensing and Communication: Joint Task Allocation and Power Optimization. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 4232–4246. [\[CrossRef\]](#)
22. Diaz-Vilor, C.; Almasi, M.A.; Abdelhady, A.M.; Celik, A.; Eltawil, A.M.; Jafarkhani, H. Sensing and Communication in UAV Cellular Networks: Design and Optimization. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 5456–5472. [\[CrossRef\]](#)
23. Ma, X.; Fan, S. Adaptive Scheduling Algorithm for AI Inference Tasks Based on Deep Reinforcement Learning in Cloud-Edge Collaborative Environment. *Ann. Appl. Sci.* **2025**, *6*, 1–29.
24. Hayat, S.; Bettstetter, C.; Fakhreddine, A.; Muzaffar, R.; Emini, D. An Experimental Evaluation of LTE-A Throughput for Drones. In *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications; DroNet'19*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 3–8. [\[CrossRef\]](#)
25. DJI. Support for Phantom 3 Professional. Available online: <https://www.dji.com/cn/support/product/phantom-3-pro> (accessed on 27 June 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.