A Deterministic Edge-AI System for Early Wildfire Smoke Detection: from Lightweight Neural Models to Operationally Reliable Surveillance

Damian Kmiecik[1,2][a] and Adrian Dziembowski[1][b]
[1]*Institute of Multimedia Telecommunications, Poznan University of Technology, Poznań, Poland*
[2]*PerfectSoft, Poznań, Poland*
{damian.kmiecik, adrian.dziembowski}@put.poznan.pl

Keywords: Wildfire Smoke Detection, Edge AI, Real-Time Embedded Systems

Abstract: Wildfire mitigation depends critically on minimizing Time-to-Detect (TTD). While lightweight convolutional neural networks (CNNs) enable visual detection of early-stage wildfire smoke, deploying them in practical, autonomous off-grid monitoring towers remains a system engineering challenge due to strict power and hardware constraints. This paper presents a proof-of-concept (PoC) implementation of a deterministic Edge-AI architecture designed for reliable wildfire surveillance. The proposed approach introduces an application-layer scheduler guaranteeing bounded system latency, as well as a hybrid approach connecting neural network output with heuristic spatio-temporal logic reducing false positives. Preliminary experimental results demonstrate that the proposed system achieves an F1-Score of 92.5% while maintaining stable latency for up to nine concurrent streams, proving that a multi-camera wildfire detection is achievable in practice, also on low-power edge hardware.

1. INTRODUCTION

Wildfires have recently evolved from sporadic environmental problem into a systemic global risk, increased by climate change, prolonged drought periods, and expanding human activity in forested regions. Regardless of the reason, the probability of successful containment of wildfire decreases exponentially with detection delay (Petrovic et al., 2012). Consequently, the main performance metric for modern wildfire surveillance systems is Time-to-Detect (TTD) (Lostanlen et al., 2025).

Traditional wildfire detection methods exhibit structural limitations. For instance, satellite-based thermal monitoring suffers from low temporal resolution (Wen and Gu, 2025), (Rajalakshmi et al., 2023) and human-operated lookouts are labor-intensive and unscalable. Therefore, automatic ground-based computer vision systems have emerged as a promising solution for early-warning systems.

Recent advances in deep learning have significantly outperformed heuristic methods based on color and motion features in smoke detection (Fofana et al, 2023). Algorithms based on lightweight architectures such as EfficientDet (Yang et al., 2022) or YOLO (Ramos et al., 2025) have demonstrated that high detection accuracy is achievable even on resource-constrained hardware, especially when evaluated against recent real-time benchmarks (Gragnaniello et al., 2023). However, there is a critical gap between model performance in laboratory settings and operational reliability in the field.

Most existing research evaluates neural models in isolation, often utilizing offline datasets and single-camera inputs. These studies frequently overlook important constraints of off-grid deployment, where systems must operate within limited power relying on solar energy. In such environments, instead of powerful GPUs, low-power edge accelerators (e.g., Edge TPU (Google, 2024)) with limited memory resources must be used.

Wildfire surveillance is a real-time embedded systems problem. In order to achieve a 360-degree coverage, a typical monitoring tower must concurrently process between 3 and 9 high resolution video streams (Alkhatib, 2024). In such scenario, naïve scaling strategies such as separate processing of

[a] https://orcid.org/0000-0003-3721-9068
[b] https://orcid.org/0000-0001-7426-3362

each stream may lead to resource contention thus non-deterministic latencies and system instability (Jain and Surve, 2020) when multiple threads compete for a single hardware accelerator. Furthermore, high per-frame accuracy does not eliminate false positives introduced by dynamic environmental factors such as fog, moving clouds, or changing illumination conditions. While temporal smoothing techniques have been proposed (Nájera De Ferrari et al., 2024), there is a lack of holistic frameworks integrating deterministic performance with practical spatio-temporal constraints.

This paper presents a conceptual architecture and a prototype implementation designed for reliable, multi-stream wildfire smoke detection. Rather than proposing a novel neural network, we focus on the system-level design required to utilize existing lightweight models effectively. The main contributions of the work are the proposal and validation of a deterministic, system-level framework for deterministic orchestration of multi-camera smoke detection, providing a scalable foundation for high-density monitoring on resource-constrained hardware. By integrating temporal aggregation, spatial consistency checks, and deterministic inference scheduling, the proposed approach stabilizes frame-level predictions and ensures predictable latency. The presented prototype demonstrates the viability of this strategy, identifying key areas for future industrial deployment. The proof-of-concept implementation demonstrates that this strategy significantly reduces false positives and effectively manages computational resources in multi-stream environments.

2. PROBLEM FORMULATION AND CONSTRAINTS

2.1 Operational Scenario and SWaP Constraints

The target environment for presented system consists of autonomous monitoring towers located in remote off-grid areas. Each observation point is equipped with a set of N fixed RGB cameras providing full 360-degree coverage of a single spot, or a set of N rotating PTZ (pan-tilt-zoom) RGB cameras providing coverage of entire area (data from multiple distinct observation towers). Typically, $N \in \{3, \dots, 9\}$. The system relies on photovoltaic energy supply, defining strict Size, Weight, and Power (SWaP) constraints. The total power budget for the computing unit is strongly limited to ensure continuous operation in variable weather conditions.

Another limitation constrains hardware resources. Typical edge platforms provide 4-16 GB of RAM (4-8 in case of microcomputers, e.g., NVIDIA Jetson or Raspberry Pi and 4-16 for typical budget PCs), no powerful GPUs, and rely on low-power co-processors (e.g., Edge TPU (Google, 2024)) for deep learning acceleration. Crucially, these accelerators typically support a single active hardware handle (Seshadri et al., 2022), creating a bottleneck for concurrent stream processing. Furthermore, passive cooling in outdoor conditions may lead to overheating, requiring temperature-aware resource management.

2.2 Functional requirements

From an operational perspective, the system must satisfy several requirements:

- Low detection latency (TTD minimization),
- Multicamera fairness (none of N video streams may be prioritized over others),
- Low false alarm rate (rare false positives to prevent operator fatigue).

Jointly, these requirements create a constrained optimization problem, as they can be met by increasing processing rate, but such an increase conflicts with the power consumption and thermal load mentioned earlier.

2.3 Software Constraints

At the software level, the system is required to operate in a general-purpose operating system (e.g., Linux-based) without reliance on custom real-time kernels. This constraint reflects commercial reality: maintainability, updateability, and compatibility with drivers – all essential for long-term operation.

However, standard OS schedulers are designed to maximize average throughput rather than provide deterministic timing or task execution order. Therefore, any real-time behavior must be enforced explicitly at the application layer through a deterministic scheduling architecture.

3. SYSTEM ARCHITECTURE AND LOGIC

The proposed solution addresses the constraints defined in Section 2 through a three-stage pipeline: a lightweight neural sensor (Section 3.1), a deterministic runtime environment (Section 3.2), and a decision layer (Section 3.3). This holistic design ensures that output of the raw model is transformed into reliable alarms within bounded time.

3.1 Lightweight Sensing Pipeline

In order to address SWaP constraints, the proposed system utilizes a quantized EfficientDet-Lite (Tan et al., 2020). object detection network. The model processes a fixed-size RGB input tensor of $D \times D \times 3$ (with $D = 320$ in the reference implementation) and outputs a confidence score $P_{raw} \in [0,1]$ and, if available, a bounding box $B = \{x_0, y_0, x_1, y_1\}$ indicating the spatial position of the detected region in normalized image coordinates.

The EfficientDet-Lite architecture undergoes full integer quantization (INT8), encompassing both model weights and activations. Such a representation allows for deterministic execution time, better thermal stability, and memory bandwidth reduction (Jafarpourmarzouni et al, 2024), being a perfect fit for a lightweight off-grid embedded system (Xiao, 2024).

Importantly, the model is not trained to distinguish various smoke types or fire intensities. The output semantics are intentionally coarse, reflecting the main objective of the system – it is an early warning mechanism, not a classification tool.

3.2 Deterministic Scheduling

The core novelty of the proposed architecture is the shift from OS-managed stochastic thread scheduler to application-level deterministic strategy. The architecture relies on three key mechanisms.

3.2.1 Buffering Latest Frame Only

Traditional video pipelines employ queues to preserve all captured frames. While such an approach is suitable for offline processing, in the proposed real-time system the minimization of latency is a crucial parameter. Therefore, traditional queues are replaced by a single mutable buffer B_c per camera stream c. This buffer always contains the most recent frame, and if a new frame arrives, it atomically overwrites the previous content. The proposed design introduces the possibility of frame drops. However, they are treated as an acceptable trade-off, as temporal continuity of proposed smoke detection is preserved by the decision logic presented in Section 3.3.

3.2.2 Mutual Exclusion

The system enforces exclusive ownership of the hardware accelerator through an explicit mutex mechanism. This is essential because the Edge TPU supports only a single active hardware handle; without a mutex, multiple concurrent stream-processing threads would cause driver-level contention and memory corruption. The mutex ensures that each inference is an atomic operation, guaranteeing system stability and predictability even under the peak load of nine camera streams.

3.2.3 Time-Division Multiplexing

The scheduler is implemented as a deterministic state machine that evaluates camera eligibility at runtime by using the binary decision function $S(t, c)$:

$$S(t,c) = \begin{cases} 1, & \text{if } (t - \tau_c \geq \Delta_t) \wedge (B_c \neq \emptyset) \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where τ_c is a timestamp indicating the last inference completion time for camera c, B_c is the buffer for camera c, and Δ_t is the minimum time gap between consecutive inferences for camera c, defined as: $\Delta_t = 1/\Gamma_{tg}$, where Γ_{tg} is the target inference frequency (e.g., 3 Hz) required to guarantee the minimum TTD.

This mechanism ensures resource fairness. Even if a camera c captures a complex scene resulting in a slightly longer inference time, the scheduler will not process camera c again until Δ_t has elapsed, ensuring that execution slots are available for other cameras.

3.3 Spatio-temporal Decision Logic

In most computer vision pipelines, the output of a neural network is treated as the final decision signal. This assumption is reasonable in offline or consumer applications where occasional classification errors are tolerable. In safety-critical monitoring systems, however, such an approach is insufficient.

Wildfire detection is not a static classification problem but a continuous perception task in a dynamic physical environment. Consequently, a single-frame predictions – regardless of their confidence – cannot be interpreted as a reliable alarm.

Therefore, the proposed system introduces a post-inference decision layer integrating outcomes of the neural network with deterministic, empiric reasoning.

This hybrid approach treats the neural network as a probabilistic sensor whose output is validated using temporal and spatial coherence constraints.

3.3.1 Detection Signal

The detection signal is modeled as a discrete-time stochastic process $\{P_t, B_t\}_{t=1}^{\infty}$, where $P_t \in [0,1]$ is the raw confidence score produced by the neural network for processed frame t of a camera stream and B_t is a corresponding bounding box of a detection.

The objective of the decision logic is to estimate a binary decision $D_t \in \{0,1\}$ representing the presence of a true smoke event, based on the observed sequence $\{P_t, B_t\}_{t=1}^{\infty}$.

3.3.2 Temporal Consistency

Empirical observations indicate that false positives are characterized by high variance and low temporal correlation, while true smoke events are characterized by stable confidence over multiple consecutive frames.

To reduce high-frequency noise in the confidence signal, an exponential moving average (EMA) filter is applied. The filtered confidence score E_t is defined recursively as:

$$E_t = \alpha P_t + (1 - \alpha) E_{t-1}\,, \tag{2}$$

where $\alpha \in (0,1)$ is the smoothing parameter. In the experiments, α was empirically selected as 0.4, providing a balance between early detection requirements and false positive suppression.

While EMA filtering reduces variance, it does not enforce the temporal consistency. To address this, the system includes a second mechanism, based on a sliding window voting technique.

Let $W_t = \{w_{t-M+1}, \dots, w_t\}$ be a binary window of length M, where:

$$w_t = \begin{cases} 1 & \text{if } E_t > C\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{3}$$

where confidence threshold$C$ is empirically set to 0.6.

A temporal consistency condition is satisfied if:

$$\sum_{k=t-M+1}^{t} w_k \geq N\,, \tag{4}$$

where $N \leq M$ is the minimum number of positive votes required. Values of $M, N,$ and C were empirically set to 5, 3, and 0.6, respectively.

This mechanism enforces temporal consistency of the system – transient spikes in input confidence cannot trigger an alarm unless they remain over multiple inference cycles.

3.3.3 Spatial Consistency

In highly dynamic scenes, the temporal consistency alone is insufficient to differentiate smoke from other temporally changing phenomena, e.g., moving clouds. Therefore, an additional step of spatial consistency analysis has been introduced.

In this step, the system evaluates the intersection over union IoU between successive bounding boxes:

$$IoU(B_{t-1}, B_t) = \frac{|B_t \cap B_{t-1}|}{|B_t \cup B_{t-1}|}. \tag{5}$$

A spatial consistency is satisfied if $IoU(B_{t-1}, B_t) > O$, where O is an overlap threshold (empirically set to 0.2).

This constraint follows the assumption that early-stage smoke exhibits gradual spatial expansion rather than abrupt spatial shift or displacement.

3.3.4 Final Decision

The final decision is produced as a conjunction of three conditions described in this section:

$$D_t = (E_t > C \ \wedge \sum_{k=t-M+1}^{t} w_k \geq N \ \wedge \ IoU(B_{t-1}, B_t) > O. \tag{6}$$

This logic transforms a noisy probabilistic signal into an interpretable decision process. Decision parameters (α, M, N, θ) were tuned to balance sensitivity with noise robustness. The sliding window $(M = 5, N = 3)$ at 3 Hz effectively filters transient spikes (e.g., insects) without delaying detection. The $IoU > 0.2$ threshold rejects fast-moving artifacts like clouds. The high mAP (0.92) and P-R curve stability (cf. Section 4) confirm system's reliability across a wide range of these threshold values.

Importantly, all thresholds can be tuned separately from the neural model, enabling adaptation to different environments without retraining.

4. EVALUATION

4.1 Model Training and Validation

In the proposed system, the EfficientDet-Lite architecture has been used. It was selected for its balance between parameter efficiency and detection performance. The used network accepts $320 \times 320 \times 3$ RGB input. The model was trained using quantization-aware training (QAT, (Jacob et al., 2018) to decrease the accuracy drop typical for post-training quantization. This process simulates low-precision effects during training, allowing the optimizer to adjust weights accordingly.

Training was conducted on 5000 high-resolution (4K) images; to ensure reproducibility despite the proprietary nature of the source data, the study strictly follows the patch-based tiling methodology and utilizes the standard EfficientDet-Lite architecture. To address the input size constraints of the EfficientDet-Lite model and to mitigate data scarcity, a patch-based tiling strategy was employed. Each image was divided into multiple overlapping fragments, generating a total of over 65,000 training samples. Fragments containing smoke were treated as positive samples, while fragments without smoke were used as negative samples, essential for reducing false positives ratio. This dataset was further enriched by a robust augmentation pipeline, including random geometric distortions and color modifications to simulate diverse weather conditions and sensor noise typical for surveillance cameras.

The training process was optimized using a total loss function L_{total} defined as:

$$L_{total} = \frac{1}{N}\sum_{i=1}^{N}(L_{cls} + \lambda L_{box}), \quad (7)$$

where L_{cls} is the Focal Loss for classification, L_{box} is the Huber Loss for bounding box regression, and the scaling parameter λ was set to 1.0. The stability and efficiency of this process are illustrated in Figure 1. The total loss function over 22,000 iterations is presented in Figure 1.

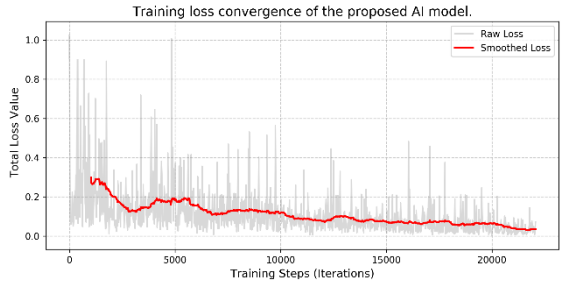

Figure 1. Training loss conergence of the proposed AI model; grey: raw loss values for each iteration, red: moving average illustrating general training trend.

As shown, the model exhibits stable convergence without significant overfitting. The stable convergence of the loss function to a value below 0.035 after 21,000 iterations confirms the successful optimization of the model weights. The absence of significant oscillations in the final training phase indicates robust generalization on validation data, providing a reliable input for the subsequent spatio-temporal decision logic.

4.2 Setup and Methodology

The proposed system was evaluated on a representative edge platform: a Raspberry Pi 4 Model B (4 GB RAM, ARM Cortex-A72), Google Coral USB Edge TPU accelerator, Linux-based operating system. This configuration properly emulates the constraints of off-grid towers, characterized by limited computing capacity and passive cooling.

To ensure repeatability of the results, camera inputs were simulated using nine prerecorded RTSP streams. The test sequences included positive samples (early-stage wildfire smoke recorded in varying lighting conditions) and hard negative samples (including fog, low clouds, dust, insects near the camera lens, and moving shadows). The simulation setup validates the architectural logic prior to field deployment. All streams were synchronized to introduce realistic contention for inference resources.

The proposed system architecture was compared against a baseline multi-process approach, where each stream is assigned to an independent OS-scheduled inference worker. The chosen baseline reflects common practice in naïve edge deployments and serves as a reference point for evaluating the benefits of the proposed approach.

System efficiency was evaluated through end-to-end latency analysis, including jitter measurements, alongside standard detection metrics such as Precision and Recall. This approach allows for a comprehensive assessment of the system's real-time processing stability under maximum load.

4.3 Latency and Stability

The temporal stability of the proposed system was measured under the maximum load ($N = 9$ cameras). The quantitative results obtained for the proposed system are presented in Tables 1 and 2.

Table 1. Efficiency of the proposed system.

Average latency	75.4 ms
Minimum latency	68.1 ms
Maximum latency	92.5 ms
Jitter	8.2 ms
System throughput	13 FPS

Table 2. Average latency of system components.

Total latency	75.4 ms
Preprocessing	50.1 ms
Inference	12.2 ms
Decision logic	13.1 ms

While the standalone inference time on Edge TPU is approximately 12 ms, the total end-to-end system latency averages at 75 ms. This increase is caused by necessary preprocessing steps (resizing high-resolution images) and the spatio-temporal decision logic performed on the host CPU.

What is crucial, the proposed deterministic scheduler allowed for keeping the latency tightly bounded – the maximum system latency did not exceed 100 ms.

4.4 Operational Reliability

4.4.1 Detection Reliability

The operational efficiency of the system was quantified using standard metrics: Precision, Recall, and F1-Score. Operational effectiveness was verified on a validation set featuring diverse scenarios,

including varying smoke densities and lighting conditions (dawn, noon, and dusk). The dataset also incorporated specific environmental distractors, such as moving clouds and fog, to rigorously test the system's robustness against false alarms. The system achieved the performance shown in Table 3.

Table 3. Efficiency of the proposed system.

Precision	Recall	F1-Score
91.3%	93.8%	92.5%

In the context of early wildfire detection, the most important metric is Recall, as it corresponds to the probability of detecting an active wildfire. A Recall higher than 93% indicates that the system successfully identifies vast majority of early wildfires, minimizing the risk of false negatives.

Simultaneously, high Precision value (> 91%) indicates that the system does not introduce high alarm fatigue connected to the necessity of false alarm handling. By effective filtering of non-smoke events, the system ensures that the majority of triggered alerts correspond to a real wildfire.

4.4.2 Spatial Accuracy

In addition to binary detection, the spatial accuracy of the predicted bounding boxes was analyzed. Figure 2 illustrates the distribution of IoU scores for all true positive detections, showing strong localization performance of the model.

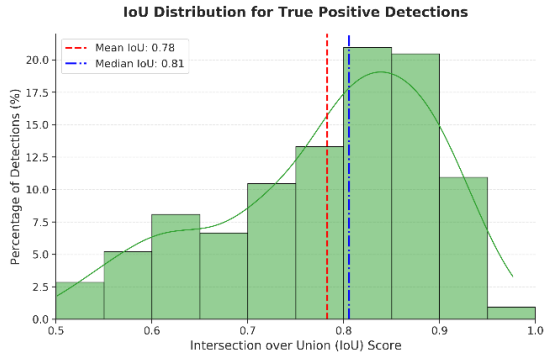

Figure 2. Distribution of Intersection over Union (IoU) scores for true positive detections. The histogram shows the frequency of IoU values together with a Kernel Density Estimation (KDE) curve.

The distribution is skewed towards higher values with median IoU of 0.81 and mean IoU of 0.78. Most of the detections fall within the [0.7, 0.9] range, well above the acceptance threshold of 0.5.

Such results prove that the proposed system not only localize the smoke, but also accurately identifies its boundaries, which is crucial for estimating the wildfire size and its expansion direction in real time.

4.5 Detection Characteristics and Operating Point

To provide an assessment of the performance beyond static metrics, a Precision-Recall (P-R) analysis was conducted. Figure 3 illustrates the P-R curve, which visualizes the trade-off between the system's ability to detect smoke (Recall) and its reliability in avoiding false alarms (Precision).

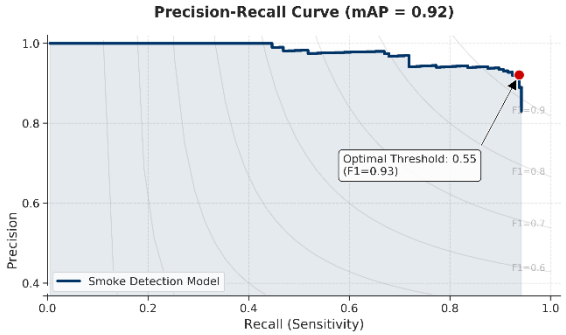

Figure 3. Precision-Recall curve with iso-F1 contours. The red marker indicates the optimal decision threshold (0.55) that maximizes the F1-score (0.93).

The model achieved a mean Average Precision (mAP) of 0.92. As presented in Figure 3, the detection curve exceeds F1 = 0.8 for almost whole sensitivity range, demonstrating high stability of the proposed approach. Crucially, the analysis identified the optimal operating point of the system (red marker in Figure 3). By setting the confidence threshold to 0.55, the system maximizes precision and recall trade-off, achieving a peak F1-score of 0.925.

The proposed Edge-AI system is competitive with more powerful solutions, based on server or GPU architectures. Crucially, over 92% F1-Score obtained on a low-power edge accelerator fully validates the hypothesis that lightweight models – when supported by deterministic spatio-temporal analysis – can meet the rigorous constraints of safety-critical surveillance.

4.6 Qualitative Analysis

To complement the extensive statistical evaluation performed on 5,000 images, we present three visual case studies illustrating system behavior in critical edge-case scenarios. These examples illustrate the interaction between the raw neural output and the post-inference decision logic.

Generalization note: to demonstrate robustness of the system, the visual examples presented in this section were obtained from external public sources and were strictly excluded from both the training and validation datasets. By successfully processing these

random, unseen sequences, we confirm that the system generalizes well to new environments and does not rely on the specific camera configurations used during its development.

Case 1: Rejection of Environmental Artifacts

Figure 4 presents a sequence of four consecutive frames, where a fast-moving cloud formation moves through the sky. Due to textural similarity, the raw neural model incorrectly classifies the cloud edge as smoke (with high confidence of 97%). However, the detected object shifts horizontally resulting in a low IoU between frames. The spatial consistency check (c.f., Section 3.3.3) detects this anomaly and decreases the output confidence to 90% for 2nd frame, 54% for 3rd frame, and below 50% for 4th one – resulting in no wildfire alarm triggered. This example demonstrates the system's ability to efficiently filter out dynamic background noise.

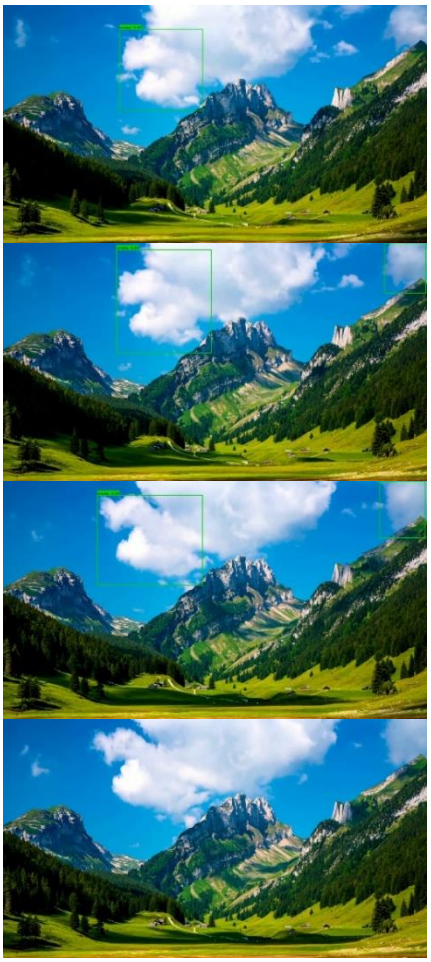

Figure 4. Sequence of 4 frames showing moving clouds.

Case 2: Confirmed Detection and Localization

Figure 5 presents a successful detection sequence of an active wildfire event. Across three consecutive frames, the system consistently detects smoke, and the bounding box remains stable, indicating high localization performance of the proposed system.

The decision logic successfully analyzes the neural network output, triggering a verified alarm with precise localization of the hazard.

Figure 5. Sequence of 3 frames showing wildfire smoke; green bounding boxes indicate a confirmed, valid detection.

Case 3: Hard Negative Failure Case (Valley Fog)

Figure 6 illustrates a False Positive triggered by a complex environmental scenario. The scene contains a mountain valley filled with dense, low clouds. The visual texture, color temperature, and slow expansion of the clouds are nearly indistiguishable from a wildfire smoke. Moreover, such clouds satisfied all temporal and spatial checks described in Section 3.3. Consequently, the system detected smoke (with 80% confidence, resulting in a triggered alarm.

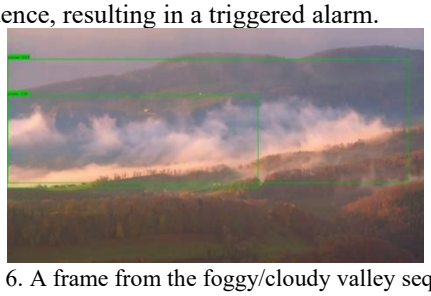

Figure 6. A frame from the foggy/cloudy valley sequence.

This case highlights inherent challenges of RGB-only surveillance systems. Scenarios with extreme visual ambiguity, such as dense valley fog, can produce signatures nearly identical to smoke in terms

of color and dynamics. While multi-modal solutions (e.g., IR) are often considered, their effectiveness in detecting early-stage, "cool" smoke at long distances remains limited. This suggests that the primary path for improvement lies not just in sensor hardware, but in refining AI models to analyze subtle gas textures, enabling better separation of meteorological phenomena from actual wildfire threats.

5. CONCLUSIONS

This paper presents a proof of concept for a deterministic Edge-AI architecture designed to reliably detect early-stage wildfire smoke under severe hardware and power constraints.

By introducing a dedicated, deterministic scheduler, the proposed system effectively resolves resource contention problems typical to multi-stream edge systems. Experimental results confirmed that the system maintains bounded latency and is able to process up to nine camera streams on a single hardware accelerator. Furthermore, the integration of a lightweight neural model with heuristic spatio-temporal logic allows for achieving an F1-Score of 92.5% without the need for powerful GPU or server-based architectures.

While the presented system successfully fills the gap between laboratory prototypes and practically deployable solutions, it remains an experimental prototype. The future work will focus on:

1. Field Validation: Transition from simulated streams to long-term field trials to assess durability in variable weather conditions.
2. System Robustness: Optimizing thermal management and power efficiency for industrial environments.
3. Edge Case Handling: Expanding the dataset by rare hard examples to improve robustness.
4. Sensory Scalability: Optional integration of additional sensors (e.g., IR) under cost and power constraints.

ACKNOWLEDGEMENTS

Publication of this article was financially supported by the Ministry of Science and Higher Education of the Republic of Poland. The research and development of the software architecture and algorithmic core described in this study were self-funded by Damian Kmiecik through his private commercial activity under the brand PerfectSoft. No public funding was utilized for the technical implementation of the prototype system.

REFERENCES

Alkhatib, A.A.A. (2024). A review on forest fire detection techniques. *Int. J. of Distrib. Sensor Networks*, vol. 9.

Fofana, T. et al. (2023). Smoke and fire detection by a convolutional neural network based on a combinatorial model. *Int. J. Innov. & Appl. Studies*, 39, pp.742-750.

Google. (2024). Coral USB Accelerator Datashet. [Online].

Gragnaniello, D. et al. (2023). Onfire contest 2023: Real-Time Fire Detection on the Edge. *Im. An. & Proc.*

Jacob, B. et al. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CVPR*, pp. 2704–2713.

Jafarpourmarzouni, S.R. et al. (2024). Enhancing Real-time Inference Performance for Time-Critical Software-Defined Vehicles. *MOST*, Dallas, USA, pp. 101-113.

Jain P.N., Surve S.K. (2020). A review on shared resource contention in multicores and its mitigating techniques. *Int. J. High Perform. Syst. Archit*. 9, 20–48.

Lostanlen, M. et al. (2025). Constructing a Real-World Benchmark for Early Wildfire Detection with the New PyroNear2025 Dataset. *arXiv:2402.05349*.

Nájera De Ferrari, F. et al. (2024). Multi-temporal assessment of a wildfire chronosequence by remote sensing, *MethodsX*, vol. 13, 103011.

Petrovic, N., Alderson, D., Carlson, J. (2012). Dynamic Resource Allocation in Disaster Response: Tradeoffs in Wildfire Suppression. *PloS one.* 7. e33285.

Rajalakshmi, R. et al. (2023). Satellite Image-Based Wildfire Detection and Alerting System Using Machine Learning. *ICDSAAI*, Chennai, India.

Ramos, L.T. et al. (2025). A study of YOLO architectures for wildfire and smoke detection in ground and aerial imagery. *Results in Engineering*, vol. 26.

Seshadri, K. et al. (2022). An Evaluation of Edge TPU Accelerators for Convolutional Neural Networks. *IISWC*, Austin, TX, USA, pp. 79-91

Tan, M. et al. (2020). EfficientDet: Scalable and efficient object detection. *CVPR.*, pp. 10781–10790.

Wen, Z., Gu, F. (2025). Wildfire Detection Based on Satellite Images Using Deep Learning. *ICAIBD*, Chengdu, China, pp. 689-694.

Xiao, P. et al. (2024). Neural Networks Integer Computation: Quantizing Convolutional Neural Networks of Inference and Training for Object Detection in Embedded Systems. *IEEE J. Sel. Top. in Appl. Earth Obs. & Remote Sens.* 17, pp. 15862-15884.

Yang, Z. et al. (2022). Smoke Detection Algorithm Based on Improved EfficientDet. *ICDLT*, NY, USA, 61–67.