Optimizing Survivor Detection Models on Thermal Imagery for UAV-Assisted Search and Rescue

Muhammed Besir Acar*, Talha Kaba*, Abdulkadir Parlak*, Ramadan Shemsu Hussen†, Ahmet Emre Gokpinar†, Selma Dilek†, *Member, IEEE* and Alperen Cakin‡
*Artificial Intelligence Engineering, Department of Computer Engineering, Hacettepe University, Ankara, Türkiye
†Department of Computer Engineering, Hacettepe University, Ankara, Türkiye
‡Graduate School of Science and Engineering, Hacettepe University, Ankara, Türkiye
Corresponding Author E-mail: *selma@ieee.org

***Abstract*—This study presents a systematic hardware-aware optimization and deployment analysis of deep neural networks for human detection in disaster areas using thermal imaging in Search and Rescue (SAR) operations. The study specifically targets real-time inference on AI-mounted UAVs, particularly the NVIDIA Jetson Orin Nano. Various state-of-the-art object detection architectures are evaluated, including YOLO models (YOLO v5–v11, v26) and transformer-based detectors (RT-DETR, RF-DETR). In 278 experimental configurations, combinations of post-training quantization (FP16/INT8 via TensorRT), unstructured pruning (30–85% sparsity), power mode scaling (7W-25W) and input resolution reduction (160–512 px) were applied. Models are trained and validated on the WiSARD wilderness thermal UAV dataset to reflect realistic SAR deployment conditions. The contributions of the project are overall comparative analysis of detection methods and a scalable/optimized deep learning framework capable of real-time survivor detection from thermal images on edge AI devices.**

***Index Terms*—Uncrewed aerial vehicles (UAVs), thermal imagery, search and rescue, survivor detection, mobile edge computing, neural network compression.**

I. Introduction

Search-and-rescue (SAR) operations in disaster-affected regions are inherently time-critical, as the probability of victim survival decreases rapidly within the first hours following an incident [1]. Timely localization of survivors is therefore essential. Traditional ground-based rescue efforts are frequently hindered by unstable debris, damaged infrastructure, limited accessibility, reduced visibility, and hazardous environmental conditions. These challenges are further exacerbated by time pressure and responder fatigue, which collectively reduce search efficiency and increase operational risk.

Uncrewed Aerial Vehicles (UAVs) have emerged as an effective technological asset for disaster response due to their rapid deployability, broad-area coverage, and ability to operate in environments that are unsafe or inaccessible to human rescuers [2]. When equipped with imaging sensors, UAVs provide aerial situational awareness and support rapid victim localization. Nevertheless, manual inspection of UAV-acquired imagery remains a significant operational bottleneck, as human operators must analyze large volumes of video streams in real time. This process is cognitively demanding and prone to fatigue-related errors, particularly during prolonged missions. These limitations have motivated the integration of automated, deep learning-based detection systems directly into UAV platforms.

979-8-3315-8150-3/26/$31.00 © 2026 IEEE

Recent advances in deep learning have led to substantial improvements in object detection performance across a wide range of applications [3], [4]. Convolutional Neural Networks (CNNs), and in particular single-stage detectors such as the YOLO family, have become the dominant choice for real-time detection in edge environments due to their efficient spatial feature extraction, favorable accuracy vs. latency trade-offs, and proven suitability for deployment on resource-constrained hardware platforms [5]. As a result, most existing UAV-based SAR systems rely on optimized CNN backbones as their primary detection architecture.

In disaster scenarios, visual sensing conditions often deteriorate rapidly due to smoke, dust, darkness, power outages, and adverse weather conditions, all of which significantly degrade the effectiveness of visible-spectrum (RGB) imaging. Thermal imaging provides a complementary sensing modality by capturing infrared radiation and highlighting temperature contrasts between humans and their surroundings. Since thermal sensors operate independently of ambient illumination, they are particularly valuable for night-time operations and low-visibility environments, including wilderness SAR missions where victims are frequently located after dark or beneath dense vegetation [6]. Nevertheless, thermal imagery introduces its own challenges: lower spatial resolution, the absence of color cues, limited texture information, increased noise levels, dynamic-range saturation, and false positives caused by environmental heat sources, all of which complicate automated detection [7]–[9]. These difficulties are compounded in UAV-based scenarios, where altitude variations, camera motion, and changing environmental conditions further degrade detection reliability [6].

In recent years, transformer-based object detection architectures have emerged as a compelling alternative to CNN-based approaches. Vision Transformers (ViTs) employ self-attention mechanisms to capture long-range contextual dependencies within images and have demonstrated strong performance in complex and cluttered scenes. Detection Transformer (DETR)

introduced an end-to-end attention-based detection framework that eliminates the need for handcrafted anchor mechanisms [10]. More recently, real-time variants such as RT-DETR have been proposed to improve computational efficiency while retaining the representational advantages of transformer-based models [11]. In SAR scenarios, where victims may be partially occluded by debris, vegetation, or thermal noise, the global context modeling capability of transformers has the potential to offer advantages over traditional CNN-based detectors. However, transformer-based detection models remain largely underexplored in the context of edge deployment for UAV-based thermal SAR applications, where embedded GPU accelerators such as NVIDIA Jetson modules typically provide only 5–30 TFLOPS of performance under tight power budgets [12]. This raises an important research question: whether modern transformer-based detectors can be practically optimized for embedded platforms while maintaining the reliability required for life-critical rescue missions.

In this study, we conduct a systematic hardware-aware evaluation of both CNN and transformer-based architectures specifically for thermal UAV-based SAR. We compare lightweight CNN detectors (i.e., the YOLO family) against real-time transformer models (RT-DETR and RF-DETR) and rigorously assess the impact of model compression techniques, namely post-training quantization, unstructured pruning, and input resolution reduction, on detection performance under real edge hardware constraints. It is important to note that we evaluate these models using WiSARD [13], a thermal UAV-SAR dataset captured in wilderness environments with realistic occlusion and varying altitudes, rather than relying on automotive or urban datasets that fail to capture the operational realities of disaster-response missions. The main contributions of this study can be summarized as follows:

1) We present a systematic comparative evaluation of CNN-based and transformer-based object detectors for thermal UAV-based SAR, revealing practical deployment trade-offs between model architecture and edge hardware constraints.
2) We provide a deployment-focused analysis of post-training quantization, unstructured pruning, and resolution reduction on the NVIDIA Jetson Orin Nano, a standard edge AI platform for UAVs.
3) We perform real-world benchmarking that characterizes accuracy-latency-memory Pareto frontiers for mission-critical performance planning.
4) We conduct an empirical investigation into the feasibility and practical limitations of transformer-based detection models for edge AI in time-critical disaster-response UAV systems.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the methodology, including the detection pipeline, baseline architectures, training protocol, and optimization techniques. Section IV presents the experimental setup and results, followed by a discussion of findings and limitations. Finally, Section V concludes the paper and outlines directions for future work.

II. Related Work

In this section, we review prior work on thermal-based human detection, edge AI optimization, and transformer-based object detection, with an emphasis on real-time performance and deployment constraints in SAR scenarios.

A. Thermal-Based Human Detection and Tracking

In [14], the authors presented a thermal image-based multi-target tracking system for SAR operations using drones, integrating YOLOv5 for human detection with a Kalman filter-based tracker that incorporates adaptive gating and track-to-track fusion. Although their approach demonstrated robust performance under occlusion and clutter in thermal environments, the dataset used was small and lacked variability. Moreover, the proposed system was not optimized for real-time edge deployment.

Dong et al. [15] combined YOLOv3-MobileNet with sensitivity-based pruning and knowledge distillation, achieving 26.6 FPS on the Jetson TX2 for thermal UAV-based SAR. However, their evaluation was limited to YOLOv3-era architectures and a small custom dataset, with no quantization or transformer comparison.

Tsai et al. [16] demonstrated that thermal imaging combined with deep learning can effectively detect humans in low-visibility environments such as smoke-filled scenarios. Their work highlights the importance of using standardized thermal imaging sensors (e.g., NFPA 1801-compliant TICs) and real-time processing pipelines for safety-critical applications in emergency scenarios.

Ulmămei et al. [17] conducted a comprehensive evaluation of object detection and semantic segmentation models for human detection in UAV thermal imagery, with a focus on mountain SAR scenarios. The authors introduced a new mountain-oriented thermal dataset combined with ten existing sources to form a benchmark of over 75,000 images. A comparative study of YOLOv8/9/10, RT-DETR, and U-Net variants was performed on an NVIDIA Jetson AGX Orin, where segmentation models, particularly Attention U-Net, outperformed detection-based approaches with an F1-score of 0.91 and up to 27 FPS in FP16 mode. RT-DETR performed poorly relative to YOLO variants on small thermal targets, which highlights the limitations of transformer-based detectors in low-contrast, data-scarce thermal domains.

B. Edge AI Optimization and Model Compression

In [18], the authors explored the optimization of CNNs for ophthalmic disease detection on resource-constrained devices. Through pruning, quantization, and knowledge distillation, they achieved a fourfold reduction in model size and significant improvements in inference speed on a Jetson Nano. In particular, quantization maintained 97% accuracy while reducing the inference time to 0.63s. Their study directly informs the optimization strategy of the current work by

offering a quantifiable trade-off framework between accuracy, latency, and model size for edge devices.

A key technique used in this study is post-training quantization, which reduces numerical precision from 32-bit floating-point (FP32) to lower bit-widths such as 16-bit (FP16) or 8-bit integers (INT8). For a weight $w \in \mathbb{R}$, quantization can be expressed as $w_q = \text{round}\left(\frac{w}{s}\right) - z$, where s is the scale factor, z is the zero-point, and w_q is the quantized weight [19]. This can reduce memory footprint by up to 4× and accelerate inference through efficient integer arithmetic. Moreover, resolution reduction decreases the spatial dimensions from $H \times W$ to $H' \times W'$ where $H' < H$ and $W' < W$. Since computational cost scales quadratically with resolution for most operations, reducing resolution by a factor of 2 can theoretically reduce FLOPs by a factor of 4:

$$\text{FLOPs}_{\text{reduced}} = \text{FLOPs}_{\text{original}} \times \left(\frac{H'}{H}\right) \times \left(\frac{W'}{W}\right) \quad (1)$$

Although these optimizations can significantly reduce inference time and memory footprint, thereby enabling deployment on resource-constrained platforms, they may incur accuracy penalties that must be carefully evaluated for mission-critical applications [20].

C. Thermal Vision Optimization for Edge Devices

Farooq et al. [21] designed an edge-deployed thermal vision system for vehicular applications, leveraging YOLOv4-based detection and thermal data preprocessing optimized for embedded hardware through quantization and asynchronous multithreading. Although effective in vehicular contexts, their work lacked adaptive scaling and on-device learning mechanisms that are essential for dynamic drone environments.

In [22], the authors proposed MS-YOLO, a lightweight YOLOv8-based framework for infrared object detection targeting real-time edge deployment. By replacing the CSPDarknet backbone with MobileNetV4 and introducing a SlideLoss function to address class imbalance and occlusion, their approach improved detection efficiency while maintaining competitive accuracy on the FLIR ADAS V2 thermal dataset. However, the study primarily focused on urban ground-based infrared scenarios rather than aerial or SAR-oriented missions.

Boddu and Mukherjee [23] further explored quantized YOLOv4-Tiny models for aerial object detection on the Raspberry Pi 5, successfully achieving real-time inference for emergency detection tasks and validating that low-bit quantization is feasible on cost-efficient hardware. Nonetheless, their system was limited to visible-spectrum imagery and did not consider thermal imaging.

Furthermore, Dinh and Kim [24] proposed a lightweight YOLOv8 model optimized with attention modules and depth-wise convolutions, demonstrating improved detection performance and processing speed for thermal images. Their findings support the notion that architectural refinements combined with quantization can enhance edge-deployed thermal detectors, an insight directly applicable to drone-based SAR missions.

D. Transformer-Based Object Detection

Transformer-based object detectors such as DETR [25] have introduced end-to-end attention-based learning that eliminates handcrafted anchors and post-processing steps by formulating detection as a direct set prediction problem using a transformer encoder-decoder architecture. The core mechanism is the scaled dot-product attention, computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right)V \quad (2)$$

where Q, K, V are the query, key, and value matrices, d_k is the dimension of the key, and T denotes the matrix transpose.

RT-DETR extends this paradigm with a hybrid encoder-decoder design in which a CNN or ViT backbone extracts multi-scale features, an efficient encoder fuses them via intra-scale and cross-scale attention, and a transformer decoder with auxiliary prediction heads performs object queries [26], [27].

Subsequent variants such as Deformable DETR [28], DINO-DETR [29], and Grounding-DINO [30] introduced multi-scale deformable attention, improved convergence, and language grounding capabilities, respectively. These advancements yielded improved contextual understanding and accuracy. However, transformer-based models are computationally intensive due to the quadratic complexity of self-attention with respect to sequence length, requiring high memory bandwidth that renders them challenging for direct edge deployment [31]. Recent hybrid approaches attempt to address this issue by combining CNN-based feature extractors with transformer attention layers or employing teacher-student distillation to transfer transformer-level performance to lightweight CNNs. These hybridization strategies serve as a conceptual foundation for the optimization process adopted in this study.

E. Summary and Research Gap

Table I presents a structured comparison of the discussed studies across key dimensions relevant to SAR deployments, including thermal sensing, UAV integration, edge hardware usage, real-time capability, and optimization strategies. As can be observed, only a limited subset of the prior work simultaneously addresses thermal imagery, UAV-based operation, and real-time inference on edge hardware.

TABLE I: Comparison table of related work with the proposed approach

Work	Year	Thermal	UAV	Edge HW	Real-Time	Optimized	Platform
[14]	2024	✓	✓		✓		N/A
[15]	2020	✓	✓	✓	✓	✓	NVIDIA Jetson TX2
[16]	2022	✓		✓	✓	✓	NVIDIA GeForce and Jetson series
[18]	2025			✓	✓	✓	NVIDIA Jetson Orin Nano
[21]	2023	✓		✓	✓	✓	Raspberry Pi 4 Model B
[22]	2025	✓				✓	N/A
[23]	2025		✓	✓	✓	✓	Raspberry Pi 5
[24]	2025	✓		✓	✓	✓	NVIDIA Jetson AGX Orin
[17]	2025	✓	✓	✓	✓	✓	NVIDIA Jetson AGX Orin
Our Work	2026	✓	✓	✓	✓	✓	NVIDIA Jetson Orin Nano

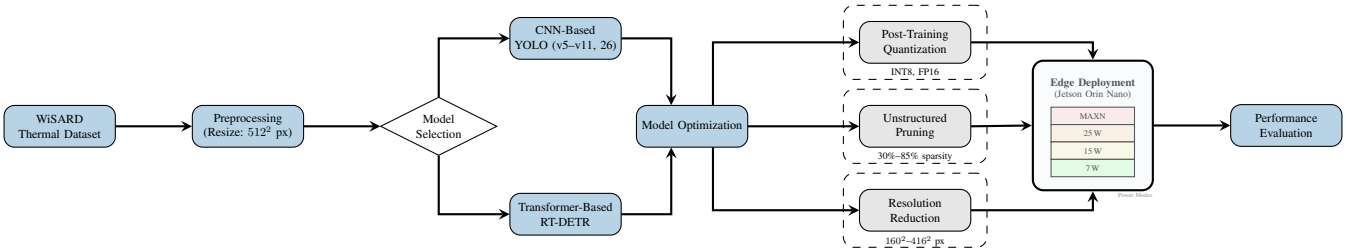


Fig. 1: End-to-end optimization and deployment flowchart for thermal survivor detection on embedded UAV platforms. The edge-deployment block shows the four evaluated power profiles of the Jetson Orin Nano.

This study directly addresses these gaps by adopting a system-level, deployment-first approach. Rather than focusing on a single architecture or optimization method, we conduct a controlled comparison of multiple YOLO-based CNNs and transformer-based detectors using a unified thermal SAR dataset, consistent metrics, and target edge hardware (i.e., NVIDIA Jetson Orin Nano). Furthermore, we advance beyond prior optimization studies by empirically validating post-training quantization (FP16 and INT8) and analyzing the practical limitations of unstructured pruning and power mode scaling on real edge hardware, aspects often overlooked in earlier research.

III. Methodology

This study proposes an extensive comparative analysis of CNN and ViT architectures for object detection, specifically human detection, in resource-constrained edge environments. Specifically, we evaluate the efficacy of post-training quantization (PTQ), unstructured pruning, power mode scaling and input dimension reduction on the trade-off between inference latency and detection accuracy.

A. System Overview

The proposed detection pipeline is designed to expedite real-time thermal object detection on Uncrewed Aerial Vehicles (UAVs). The workflow (illustrated in Fig. 1) covers data ingestion, data preprocessing, model inference, and performance evaluation. The primary data source is the WiSARD thermal imaging dataset, collected via UAV platforms under diverse environmental conditions [13]. The dataset captures varying occlusion scenarios, weather patterns, and geographical settings, providing a robust testbed for thermal detection. Since the source data is already high quality, we keep preprocessing to a minimum to save computing resources. The raw thermal images undergo resizing to varying input resolutions (160×160, 320×320, 416×416, and 512×512 pixels) to assess the impact of spatial resolution on model throughput. Following preprocessing, the imagery is fed into the inference engine hosting the selected YOLO and Detection Transformer models. We deploy these models on two distinct hardware configurations: a high-performance workstation for baseline training and a specific edge device, the NVIDIA Jetson Orin Nano.

B. Baseline Detection Models

To provide a comprehensive evaluation of current state-of-the-art architectures, we select representative models from two distinct paradigms: CNN-based detectors and Transformer-based detectors.

1) YOLO (You Only Look Once): The YOLO family of models represents the state-of-the-art in one-stage CNN-based object detection. To capture a broad range of architectural evolution, we evaluate models spanning from v5 to v26. Specific variants: nano (n), tiny (t), and small (s) were chosen to target the low-latency requirements of edge deployment. YOLO architectures typically consist of a Backbone (for feature extraction), a Neck (for feature aggregation across scales), and a Head (for predicting class probabilities and bounding box coordinates). They frame object detection as a regression problem, processing the entire image in a single forward pass, which makes them quite efficient.

2) RT-DETR (Real-Time Detection Transformer) and RF-DETR: Unlike CNNs, which operate on inductive biases such as translation invariance, ViTs prioritize self-attention to model long-range dependencies within an image. We employ both RT-DETR [26] and RF-DETR [32] as representative architectures of this family. RT-DETR overcomes the slow convergence of earlier DETR architectures by employing a hybrid encoder and IoU-aware query selection, which allows global context understanding without sacrificing the speed advantages associated with CNNs. RF-DETR extends this further by incorporating a self-supervised ViT backbone and deformable attention mechanisms, which eliminate the need for anchor boxes and Non-Maximum Suppression.

C. Training Protocol

All baseline models were trained using the Ultralytics framework to ensure a standardized experimental environment. The training regimen consisted of 20 epochs for YOLOs and 72 for ViTs with an early stopping patience of 10 epochs to prevent overfitting on the thermal dataset. A consistent batch size of 16 was maintained across all experiments. To establish a stable, high-precision baseline prior to quantization, Automatic Mixed Precision (AMP) was explicitly disabled (amp=False), enforcing full FP32 precision during the training phase. The optimizer configuration was set to auto, which consistently selected the AdamW optimizer. Default Ultralytics learning rate schedules and hyperparameters were retained to

ensure reproducibility and fair comparison across the diverse model architectures.

D. Optimization Techniques

To tailor these complex architectures for edge deployment, we apply a series of model compression techniques targeted to reduce memory footprint and Floating Point Operations (FLOPs).

1) Post-Training Quantization (PTQ) via TensorRT: We harness NVIDIA TensorRT to perform Post-Training Quantization (PTQ). Models were converted from their native PyTorch format (`.pt`) to the highly optimized `.engine` format. This process includes fusing layers and selecting optimal kernel implementations specific to the target GPU architecture. We evaluate three precision levels:

- FP32 (Single Precision): Used as the uncompressed baseline for accuracy reference.
- FP16 (Half Precision): Reduces memory usage while typically maintaining near-baseline accuracy.
- INT8 (8-bit Integer): Drastically reduces memory bandwidth usage and leverages specialized Tensor Cores on the Orin Nano for maximum throughput.

Crucially, the quantization and engine generation were performed directly on the target hardware to make sure the calibration process correctly accounted for the specific GPU architecture.

2) Unstructured Pruning: We employ unstructured pruning to systematically remove redundant network parameters. By using L1 and L2 norm magnitude pruning, we target individual weights with the lowest contributions to the network's output. We experimented with five distinct sparsity levels: 30%, 55%, 75%, 80%, and 85%, to rigorously map the degradation curve of accuracy versus parameter count. Structured pruning (removing entire channels) was not utilized due to current limitations in the Ultralytics framework's support for this operation on the target architectures.

3) Input Dimension Reduction: We systematically evaluate model performance across a spectrum of input resolutions (160px to 512px). Lowering the input resolution quadratically reduces the pixel count and directly lowers inference FLOPs. However, this efficiency may come at the expense of detecting small thermal signatures.

E. Evaluation Metrics

The models are evaluated using standard COCO metrics, with a primary focus on Mean Average Precision (mAP_{50} and $mAP_{50:95}$), Precision, Recall, and F1-score. Operational efficiency is assessed via inference latency (ms), total parameter count, and model size (MB).

IV. Experimental Results

A total of 14 detector variants spanning six architectural generations are evaluated as baselines: YOLOv5 (n, s), YOLOv8 (n, s), YOLOv9 (t, s), YOLOv10 (n, s), YOLO11 (n, s), YOLO26 (n, s), RT-DETR, and RF-DETR. The nano (n) and tiny (t) variants target the lowest-latency deployment tier; small (s) variants offer the next accuracy tier. The two transformer-based detectors (RT-DETR, RF-DETR) are included to empirically assess the feasibility of attention-based architectures under hard edge hardware constraints. Across five optimization categories: baseline (FP32), post-training quantization (FP16, INT8), input resolution reduction (160–512 px), unstructured pruning (L1/L2, 30–85% sparsity), and combined quantization + resolution reduction, a total of 278 experiment configurations are evaluated across both hardware platforms.

A. Experimental Setup

All baseline training was conducted using the NVIDIA Tesla T4 GPU available through Google Colab, equipped with 16 GB of GDDR6 memory, along with 12.7 GB of system RAM. The T4 serves as the reference training and benchmarking platform for FP32 baseline measurements. Edge deployment experiments were conducted on an NVIDIA Jetson Orin Nano (8 GB LPDDR5) development kit, which features a 6-core Arm Cortex-A78AE CPU clocked at up to 1.5 GHz, a 1024-core NVIDIA Ampere GPU with 32 Tensor Cores, and a sustained TDP of 10 W. Persistent storage is provided by a PNY CS2230 M.2 NVMe SSD (PCIe 3.0 x4, 500 GB).

All models excluding RF-DETR were implemented and trained using the Ultralytics framework (v[8.4.2x]) with PyTorch [2.6.0]. YOLO models were trained for a maximum of 20 epochs, while ViT-based models (RT-DETR and RF-DETR) were trained for a maximum of 72 epochs to accommodate their characteristically slower convergence. Early stopping with a patience of 10 epochs was applied across all models to prevent overfitting on the limited WiSARD thermal subset. A fixed batch size of 16 was maintained. Automatic Mixed Precision (AMP) was explicitly disabled (amp=False) to enforce full FP32 precision during training and establish a clean baseline prior to quantization. The optimizer was set to auto, which consistently selected AdamW across all models.

B. Datasets

1) Dataset Selection and Comparison: To ensure the proposed models are robust to the operational realities of aerial Search and Rescue (SAR), we evaluated several standard thermal object detection benchmarks (see Table II).

- FLIR ADAS: While popular, this dataset is collected from a ground-vehicle perspective in urban environments. The geometric features of targets (pedestrians on sidewalks) and backgrounds (buildings, paved roads) do not generalize to the high-altitude, top-down look-angles of UAVs.
- KAIST Multispectral: Similarly, KAIST focuses on urban pedestrian detection from a vehicle's point of view, lacking the unstructured terrain and occlusion patterns typical of wilderness scenarios.

Consequently, we utilized the WiSARD (Wilderness Search and Rescue Dataset). This dataset is uniquely aligned with our mission profile, offering high-fidelity thermal imagery captured specifically from UAVs across diverse, unstructured wilderness environments. It provides the critical "domain

TABLE II: Comparison between public thermal/RGB datasets.

Dataset	Year	Size	Boxes	Task	Platform	Sensor	Alt. (m)	Occl.	Object Categories
Campus [33]	2016	930k	11m	OT	UAV	RGB	80	-	Pedestrian, bike, car, bus, etc.
VisDrone [34]	2018	40k	2.5m	OD, OT	UAV	RGB	-	-	Pedestrian, vehicle, bicycle, tricycle
BIRDSAI [35]	2020	62k	154k	WOD	UAV	LWIR	60-120	-	Person, animal
NII-CU [36]	2020	5.9k	19k	OD	UAV	RGB+LWIR	20-50	-	Person
AOS [37]	2020	4.3k	23k	OD	UAV	LWIR	30-35	✓	Person
KAIST [38]	2021	95k	103k	OD	GV	RGB+LWIR	0	✓	Person, people
UMA-SAR [39]	2021	309k	615k	SAR	UGV	RGB+LWIR	0	-	Person, vehicle, debris
WiSARD (Used)	**2022**	**56k**	**74k**	**SAR**	**UAV**	**RGB+LWIR**	**20-120**	✓	**Person**
FLIR ADAS [40]	2022	26k	344k	OD	GV	LWIR	0	-	Person, bike, car, bus, etc.
HIT-UAV [41]	2023	2.9k	43k	OD	UAV	LWIR	60-130	-	Person, vehicle, bicycle

OD: Object Detection, OT: Object Tracking, SAR: Search and Rescue, WOD: Wildlife Object Detection, GV: Ground Vehicle, UGV: Uncrewed Ground Vehicle, UAV: Uncrewed Aerial Vehicle, LWIR: Long Wave Infrared, RGB: Red-Green-Blue.

shift" challenges, such as heavy foliage occlusion, small target signatures (high altitude), and varying thermal crossover periods (dawn/dusk), which are absent in automotive datasets.

2) WiSARD Dataset Characteristics: The experimental validation relies on the thermal subset of WiSARD, utilizing the Long-Wave Infrared (LWIR) imagery.

- Volume: The specific subset utilized comprises 3658 labeled thermal images.
- Sensor Specifications: Images were acquired using varying UAV platforms, typically offering a native resolution of 640×512 pixels, which is standard for uncooled microbolometer thermal sensors.
- Scenarios: The data spans robust environmental diversity, including dense coniferous forests, coastal regions, and snow-transition zones, ensuring the models do not overfit to a single background texture.

3) Preprocessing and Partitioning:

- Annotation Format: We converted ground truth labels to the standard YOLO format (normalized center-x, center-y, width, height) to ensure compatibility with the Ultralytics API.
- Data Splits: To rigorously evaluate generalization, we partitioned the dataset into strict Training (80%), Validation (10%), and Testing (10%) subsets.
- Preprocessing: Given the high signal-to-noise ratio of the WiSARD thermal data, extensive filtering was deemed unnecessary. The primary preprocessing step involved resizing the raw thermal frames to the target input resolutions (160px – 512px) discussed in the methodology, utilizing bilinear interpolation to preserve small thermal hotspots.

C. Results

A total of 278 experiment configurations were evaluated across four edge device power profiles (7W, 15W, 25W, and MAXN), spanning 14 detector variants across six optimization axes: FP32 baselines, TensorRT FP16/INT8 quantization, input resolution reduction (160–512 px), L1/L2 unstructured pruning (30–85% sparsity), and combined quantization + resolution reduction.

1) Baseline Performance (FP32, 512 px): Table III reports the FP32 baselines on the Jetson Orin Nano at 7 W, sorted by $mAP_{50:95}$. RT-DETR achieves the highest overall accuracy with $mAP_{50} = 0.966$ and $mAP_{50:95} = 0.629$ at 57.3 ms latency ($\sim$17 FPS), though its 63.1 MB footprint and relatively high latency make it less suitable for resource-constrained deployment. RF-DETR ranks second in $mAP_{50:95}$ (0.549) with a competitive mAP_{50} of 0.937, but its 127.5 MB model size and 81.9 ms latency impose a considerable cost for marginal accuracy gains over lighter alternatives. Among CNN-based detectors, YOLO26s delivers the best accuracy with $mAP_{50} = 0.952$ and $mAP_{50:95} = 0.548$ at only 15.8 ms, making it the strongest compact option. YOLOv5n stands out for efficiency, achieving $mAP_{50} = 0.938$ and $mAP_{50:95} = 0.536$ at just 8.2 ms with a 5.0 MB footprint, offering the best latency–accuracy trade-off in this benchmark. Nano-scale models such as YOLO11n, YOLO26n, and YOLOv8n cluster around $mAP_{50} \approx 0.926$–0.928 with sub-9 ms latencies, confirming that highly compact architectures remain viable for real-time SAR detection on the Orin Nano.

TABLE III: FP32 baseline performance on Jetson Orin Nano (7 W, 512 px). Models sorted by $mAP_{50:95}$.

Model	Size	mAP_{50}	$mAP_{50:95}$	Recall	Lat. (ms)
RT-DETR	63.1 MB	0.9660	0.6290	0.9500	57.3
RF-DETR	127.5 MB	0.9372	0.5488	0.9474	81.9
YOLO26s	19.4 MB	0.9518	0.5481	0.9205	15.8
YOLOv5n	5.0 MB	0.9382	0.5363	0.8975	8.2
YOLOv9t	4.4 MB	0.9263	0.5359	0.8854	11.4
YOLOv5s	17.6 MB	0.9247	0.5308	0.8892	13.9
YOLOv9s	14.5 MB	0.9226	0.5289	0.8808	18.9
YOLO11n	5.2 MB	0.9275	0.5247	0.8855	8.4
YOLO11s	18.3 MB	0.9276	0.5236	0.8717	15.0
YOLO26n	5.1 MB	0.9249	0.5228	0.8854	8.6
YOLOv10s	15.7 MB	0.9122	0.5069	0.8598	17.5
YOLOv8s	21.5 MB	0.9116	0.5013	0.8719	14.1
YOLOv8n	6.0 MB	0.9263	0.4995	0.8679	7.6
YOLOv10n	5.5 MB	0.8927	0.4877	0.8303	8.6

2) TensorRT Quantization: Table IV presents the impact of TensorRT quantization on the Jetson Orin Nano at 7 W. RT-DETR-FP16 achieves the highest accuracy with $mAP_{50} = 0.967$ at 29.7 ms, a roughly $2\times$ speedup over its

FP32 baseline while marginally improving accuracy by 0.1 pp, likely due to TensorRT's layer fusion and kernel optimization. RF-DETR-FP16 yields a competitive mAP_{50} of 0.937 at 76.8 ms; however, its INT8 variant, despite a marginal accuracy gain, incurs an extreme latency of nearly 11 s per frame, rendering it entirely infeasible for real-time deployment and thus excluded from further consideration.

Among YOLO models, FP16 quantization consistently delivers reliable accuracy retention. YOLO26s-FP16 stands out with $mAP_{50} = 0.955$ at 13.6 ms, actually *improving* over its FP32 baseline by 0.3 pp. Similarly, YOLO26n-FP16, YOLOv5s-FP16, and YOLOv10s-FP16 each exhibit small positive Δ values, suggesting that TensorRT's calibration and kernel optimizations can act as an implicit regularizer for certain architectures.

INT8 quantization, however, presents a more mixed picture. While YOLO11s-INT8 and YOLOv5s-INT8 maintain competitive accuracy with mAP_{50} drops of only 0.4 pp and 0.5 pp respectively, several models suffer severe degradation: YOLOv5n-INT8 loses 10.6 pp, YOLOv9t-INT8 loses 10.5 pp, and YOLO11n-INT8 loses 8.9 pp relative to their FP32 baselines, indicating that these compact architectures are particularly sensitive to aggressive quantization. YOLOv10 and 26 variants were only convertible to FP16 engines; INT8 conversion was not supported by the current TensorRT runtime for these architectures. Overall, PTQ via TensorRT is the single most impactful optimization for Orin Nano deployment.

TABLE IV: TensorRT quantization on Jetson Orin Nano (7 W, 512 px). Δ is mAP_{50} change vs. FP32 baseline. Models sorted by mAP_{50}.

Model	Prec.	mAP_{50}	Δ	Lat. (ms)	Size
RT-DETR	FP16	0.9670	+0.0010	29.7	66.1 MB
RF-DETR	INT8	0.9375	+0.0003	10956.3	44.8 MB
RF-DETR	FP16	0.9372	+0.0000	76.8	63.8 MB
YOLO26s	FP16	0.9549	+0.0031	13.6	21.8 MB
YOLOv5n	FP16	0.9339	−0.0043	7.5	7.3 MB
YOLO11s	FP16	0.9269	−0.0007	13.7	21.7 MB
YOLO26n	FP16	0.9265	+0.0016	9.1	7.7 MB
YOLOv5s	FP16	0.9249	+0.0002	11.4	20.5 MB
YOLOv8n	FP16	0.9247	−0.0016	8.0	8.4 MB
YOLO11n	FP16	0.9240	−0.0035	8.8	8.3 MB
YOLO11s	INT8	0.9235	−0.0041	9.6	12.0 MB
YOLOv9t	FP16	0.9218	−0.0045	12.8	6.8 MB
YOLOv9s	FP16	0.9208	−0.0018	18.1	17.1 MB
YOLOv5s	INT8	0.9194	−0.0053	8.3	10.8 MB
YOLOv10s	FP16	0.9151	+0.0029	14.2	16.8 MB
YOLOv8s	FP16	0.9102	−0.0014	12.7	24.1 MB
YOLOv8s	INT8	0.8951	−0.0165	8.7	12.6 MB
YOLOv10n	FP16	0.8910	−0.0017	9.2	7.1 MB
YOLOv8n	INT8	0.8815	−0.0448	6.4	4.6 MB
YOLO11n	INT8	0.8384	−0.0891	7.3	5.2 MB
YOLOv9s	INT8	0.8337	−0.0889	13.7	14.4 MB
YOLOv5n	INT8	0.8320	−0.1062	6.3	4.3 MB
YOLOv9t	INT8	0.8214	−0.1049	12.3	5.7 MB

3) Input Resolution Reduction: Fig. 2 illustrates the effect of input resolution on accuracy and latency. For YOLO models, latency remains relatively flat across resolutions (ranging from 15.78 ms at 512 px to 4.65 ms at 160 px for YOLO26s) because the Orin Nano's GPU is underutilized at smaller tensor sizes. However, mAP_{50} degrades sharply below 320 px (e.g., YOLOv5n: 0.938 → 0.672 from Base to 160 px), reflecting the loss of small thermal signatures. The sweet spot is 320 px, which preserves >87% mAP_{50} across all architectures while halving the pixel count relative to 512 px. Interestingly, RT-DETR achieves its peak accuracy at 416 px ($mAP_{50} = 0.975$) rather than at its base resolution (0.966), suggesting that the transformer's global attention mechanism may benefit from a more compact spatial representation that reduces background noise in thermal scenes.

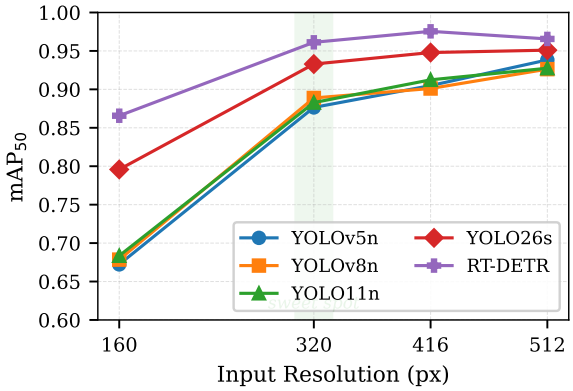


Fig. 2: Impact of input resolution on mAP_{50} (FP32 baseline, Orin Nano 7 W). All YOLO models degrade sharply below 320 px; RT-DETR peaks at 416 px rather than 512 px.

4) Unstructured Pruning: L1 and L2 magnitude-based unstructured pruning was applied to all YOLO models at 30%, 55%, 75%, 80%, and 85% sparsity. At 30% sparsity, the median mAP_{50} loss is below 0.3 pp; however, sensitivity varies significantly across architectures, with YOLOv5n exhibiting up to 2.8 pp degradation while YOLO11n shows negligible loss. Because unstructured sparsity yields irregular memory access patterns that PyTorch cannot exploit at the kernel level, *no measurable latency reduction* was observed on the Orin Nano. Beyond 55% sparsity, accuracy collapses for nano-class models: YOLO11n drops from 0.928 → 0.786 at 75% and to 0.559 at 80% (see Fig. 3). Larger "small" variants demonstrate notably better resilience: YOLOv9s retains 0.908 mAP_{50} at 75% sparsity (−1.4 pp), and YOLO11s retains 0.890 (−3.8 pp). L1 and L2 norms produce near-identical results (<0.2 pp difference). These findings indicate that unstructured pruning is ineffective as a standalone edge-optimization strategy; structured pruning or sparsity-aware runtimes would be required to convert parameter reduction into latency gains.

5) Power Mode Analysis: The Orin Nano supports four power profiles (7 W, 15 W, 25 W, MAXN). Accuracy remains unchanged across all modes for every configuration, confirming that power-mode scaling affects latency rather than detection quality. Table V reports latency for the three Pareto-optimal TensorRT engines. The 15 W profile provides no meaningful improvement over 7 W, whereas 25 W captures

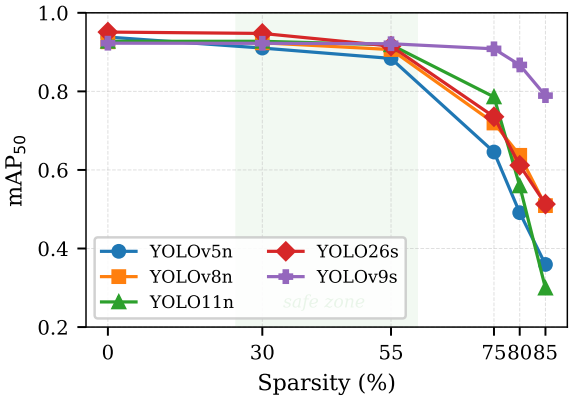


Fig. 3: Unstructured pruning (L1) impact on mAP_{50} (FP32, 512 px). Nano-class models collapse beyond 55% sparsity, while "small" variants (YOLO26s, YOLOv9s) retain higher accuracy at equivalent sparsity levels.

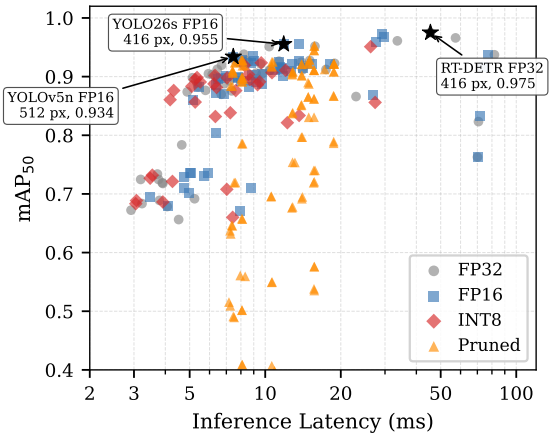


Fig. 4: Accuracy vs. latency Pareto frontier on Jetson Orin Nano (7 W). Annotated points mark the three recommended deployment tiers.

most of the achievable speedup. MAXN yields the lowest latency overall, but only a modest additional gain over 25 W (e.g., RT-DETR FP32 416px: 102.2 → 85.4 ms). For battery-powered UAV deployment, these results make 7 W the most practical default operating point, while 25 W/MAXN are better suited to latency-critical missions.

From an energy perspective, the 7 W mode is significantly more efficient: YOLOv5n-FP16 consumes approximately 25.2 mJ per inference at 7 W (3.6 ms × 7 W) versus 60 mJ at MAXN (2.4 ms × 25 W), representing a 2.4× energy advantage despite only 1.5× higher latency. This tradeoff strongly favors 7 W for battery-constrained UAV missions where flight endurance is the primary constraint.

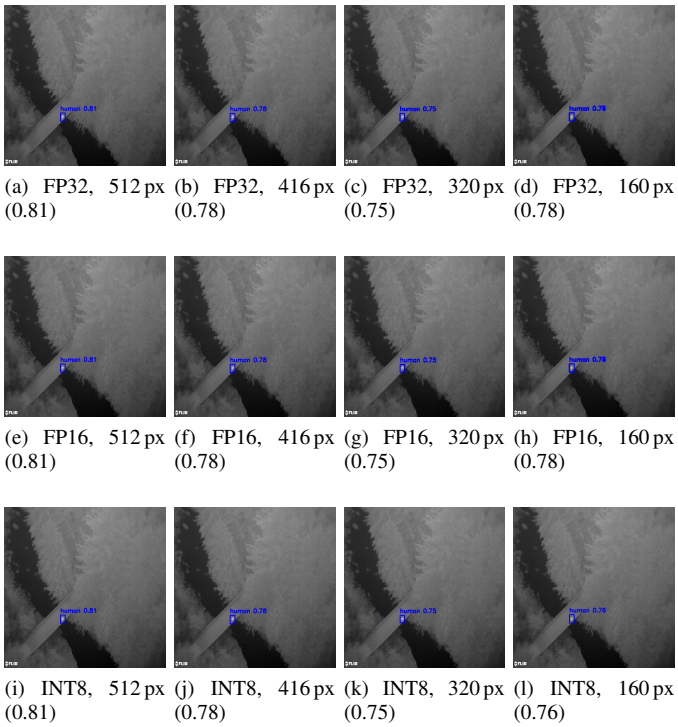
TABLE V: Latency (ms) across Orin Nano power modes for Pareto-optimal TensorRT engines.

Engine	7 W	15 W	25 W	MAXN	Speedup
RT-DETR FP32 416px	110.1	110.2	102.2	85.4	1.29×
YOLO26s FP16 416px	5.9	5.9	4.3	3.9	1.51×
YOLOv5n FP16 512px	3.6	3.6	2.6	2.4	1.50×

6) Pareto-Optimal Configurations: Fig. 4 presents the Pareto frontier across all configurations. Three deployment tiers emerge:

- **Maximum accuracy:** RT-DETR FP32 at 416 px achieves $mAP_{50} = 0.975$ at 45.5 ms (∼22 FPS), suitable for non-time-critical post-flight analysis.
- **Balanced:** YOLO26s-FP16 at 416 px delivers $mAP_{50} = 0.955$ at 11.85 ms (∼84 FPS), offering near-peak accuracy at real-time speed.
- **Ultra-low latency:** YOLOv5n-FP16 at 512 px reaches $mAP_{50} = 0.934$ at 7.46 ms (∼134 FPS), enabling multi-stream or battery-critical scenarios with only 7.3 MB engine size.

When quantization is combined with resolution reduction, a fourth *ultra-efficient* tier emerges: YOLO26s-FP16 at 416 px achieves $mAP_{50} = 0.955$ at 11.85 ms (84 FPS), delivering near-balanced-tier accuracy at ultra-low-latency-tier speed with only 21.4 MB engine size. This configuration is particularly attractive for multi-camera UAV platforms requiring simultaneous inference on multiple thermal streams.



(a) FP32, 512 px (0.81) (b) FP32, 416 px (0.78) (c) FP32, 320 px (0.75) (d) FP32, 160 px (0.78)

(e) FP16, 512 px (0.81) (f) FP16, 416 px (0.78) (g) FP16, 320 px (0.75) (h) FP16, 160 px (0.78)

(i) INT8, 512 px (0.81) (j) INT8, 416 px (0.78) (k) INT8, 320 px (0.75) (l) INT8, 160 px (0.76)

Fig. 5: RF-DETR detection on a WiSARD test frame. Top row: FP32; middle row: TensorRT FP16; bottom row: TensorRT INT8. Confidence is driven by input resolution (0.81 at 512 px → 0.75 at 320 px), while quantization produces no visible change within each resolution tier (≤0.02 pp difference).

7) Qualitative Detection Examples: Fig. 5 presents RF-DETR detection outputs on a representative WiSARD test frame across four input resolutions and three precision modes (FP32, FP16, and INT8). The model correctly localises the target in all twelve configurations. Confidence scores are governed primarily by input resolution, decreasing from 0.81 at 512 px to 0.75 at 320 px, while quantization has negligible impact: FP32, FP16, and INT8 outputs are visually indistinguishable within each resolution tier, with confidence differences of at most 0.02 pp. Even at 160 px the detector maintains a valid prediction (confidence ≥ 0.76), indicating that the thermal signature remains sufficiently salient on this high-contrast frame.

D. Discussion and Limitations

In this subsection, we synthesize the key findings from our experiments, identify the most effective optimization strategies for edge-deployed thermal SAR, and discuss the limitations of the current study.

Quantization dominates. TensorRT PTQ delivers up to 2× latency reduction (excluding RF-DETR INT8) with $<$2.1 pp mAP_{50} loss on average (mean INT8 vs. FP16 gap: 5 pp), making it the single most effective optimization. This is consistent with findings in general edge-deployment literature but particularly impactful on thermal imagery, where the relatively low scene complexity (single-class, limited texture) tolerates aggressive quantization. Notably, YOLO26s-FP16 exhibits a 0.3 pp *improvement* over its FP32 baseline, suggesting that FP16 calibration may regularize certain model architectures.

Pruning without hardware support is futile. Unstructured pruning preserves accuracy at low sparsity for most architectures but yields zero latency benefit on the Orin Nano, confirming that sparse matrix support (e.g., NVIDIA's 2:4 structured sparsity or dedicated sparse tensor cores) is a prerequisite for pruning-based acceleration on edge GPUs. Pruning tolerance varies dramatically by architecture: "small" variants (YOLOv9s, YOLO11s) retain $>$90% mAP_{50} at 75% sparsity, while nano-class models (YOLOv5n, YOLO26n) collapse at the same level.

Resolution is a second-order knob. Reducing input size from 512 $\rightarrow$ 320 px is viable when combined with quantization (4–9 pp mAP_{50} loss), but dropping below 320 px causes disproportionate accuracy collapse, especially for small or occluded thermal targets characteristic of high-altitude SAR. The combined 416 px + FP16 configuration of YOLO26s yields an attractive $mAP_{50} = 0.955$ at only 11.85 ms.

Transformer viability on edge. Post-quantization, RT-DETR's 29.7 ms and RF-DETR FP16's 76.8 ms latency is competitive with heavier YOLO variants (e.g YOLOv9s: 18.9 ms FP32). Additionally, RT-DETR achieves its peak mAP_{50} at 416 px rather than 512 px, suggesting potential for further latency reduction without accuracy penalty. INT8 quantization of RF-DETR on the other hand, exploded the latency primarily because of the employed quantization approach.

Limitations. (1) Only single-class (person) detection was evaluated; multi-class SAR scenarios (e.g., vehicles, debris) may alter the accuracy–efficiency trade-off. (2) Unstructured pruning was tested without fine-tuning; iterative pruning-retraining cycles may recover more accuracy. (3) Power consumption was measured at the board level via NVPMODEL profiles rather than per-inference energy metering. (4) The WiSARD dataset, while well-suited to aerial SAR, has limited scale ($\sim$4 k images), and generalization to larger datasets remains to be validated. (5) YOLOv10 and 26 architectures lacked INT8 TensorRT support, preventing a full quantization comparison for these models.

V. Conclusion

This paper presented a comprehensive edge-deployment study of both CNN-based and transformer-based thermal object detectors for UAV search-and-rescue scenarios on the NVIDIA Jetson Orin Nano. We systematically evaluated 14 detector variants under multiple optimization strategies, including TensorRT post-training quantization, input resolution reduction, unstructured pruning, and power-mode scaling, to characterize the trade-off between detection accuracy and inference efficiency. The results showed that TensorRT quantization is the most effective deployment optimization, delivering up to 2× latency reductions with a mean accuracy loss of only 2.1 pp, while reducing the input resolution to 320 px provides a practical secondary trade-off. Among the evaluated models, RT-DETR FP16 416px achieved the highest accuracy ($mAP_{50} = 0.975$), whereas YOLO26s FP16 416px and YOLOv5n FP16 512px emerged as the most practical real-time deployment choices, offering strong accuracy at very low latency on low-power embedded hardware. Unstructured pruning, while preserving accuracy at low sparsity, yielded no latency benefit without dedicated sparse-matrix hardware support, and pruning tolerance was found to vary dramatically across architectures. From an energy perspective, the 7 W power mode offers 2.4× better energy efficiency than MAXN, making it the preferred setting for battery-constrained UAV missions. Overall, the findings demonstrate that reliable thermal human detection for UAV-based SAR is feasible on resource-constrained platforms when architecture selection and hardware-aware optimization are jointly considered. Future work will extend this framework toward multi-class SAR settings, structured sparsity, and more autonomous energy-aware onboard perception.

Acknowledgment

This study was supported by Hacettepe Scientific Research Projects (BAP) Coordination Unit under grant number 22892 and the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under grant number 123R075.

References

[1] World Health Organization, "Who guidance on emergency medical teams: Earthquake response," World Health Organization, Tech. Rep., 2021. [Online]. Available: https://iris.who.int/bitstream/handle/10665/350838/9789240037182-eng.pdf

[2] M. Erdelj and E. Natalizio, "Uav-assisted disaster management: Applications and open issues," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, pp. 1–5.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[6] P. Rudol and P. Doherty, "Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery," in *2008 IEEE Aerospace Conference*, 2008, pp. 1–8.

[7] Y. Dash, V. Gupta, A. Abraham, and S. Chandna, "Improving object detection in high-altitude infrared thermal images using magnitude-based pruning and non-maximum suppression," *Journal of Imaging*, vol. 11, no. 3, p. 69, February 2025.

[8] K. Zhou, A. Paiement, and M. Mirmehdi, "Detecting humans in rgb-d data with cnns," 2022. [Online]. Available: https://arxiv.org/abs/2207.08064

[9] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1740–1747.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020. [Online]. Available: https://arxiv.org/abs/2005.12872

[11] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," 2024. [Online]. Available: https://arxiv.org/abs/2304.08069

[12] NVIDIA Corporation, "Nvidia jetson: The ai platform for autonomous machines," 2019. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

[13] D. Broyles, C. R. Hayner, and K. Leung, "Wisard: A labeled visual and thermal image dataset for wilderness search and rescue," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2022, p. 9467–9474. [Online]. Available: http://dx.doi.org/10.1109/IROS47612.2022.9981298

[14] S. Yeom, "Thermal image tracking for search and rescue missions with a drone," *Drones*, vol. 8, no. 2, 2024. [Online]. Available: https://www.mdpi.com/2504-446X/8/2/53

[15] J. Dong, K. Ota, and M. Dong, "Real-time survivor detection in uav thermal imagery based on deep learning," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, 2020, pp. 352–359.

[16] P.-F. Tsai, C.-H. Liao, and S.-M. Yuan, "Using deep learning with thermal imaging for human detection in heavy smoke scenarios," *Sensors*, vol. 22, p. 5351, 07 2022.

[17] A.-A. Ulmămei, T. D'Adamo, C.-E. Vasile, and R. Hobincu, "Human detection in uav thermal imagery: Dataset extension and comparative evaluation on embedded platforms," *Journal of Imaging*, vol. 11, no. 12, 2025. [Online]. Available: https://www.mdpi.com/2313-433X/11/12/436

[18] Z. Abushaban, S. E. Yüzbaşıoğlu, S. Dilek, and S. Tosun, "Optimizing deep learning models for ophthalmic disease detection on resource-constrained devices," in *2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA)*, 2025, pp. 1–9.

[19] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," 2017. [Online]. Available: https://arxiv.org/abs/1712.05877

[20] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," 2021. [Online]. Available: https://arxiv.org/abs/2106.08295

[21] M. A. Farooq, W. Shariff, F. Khan, and P. Corcoran, "Development, optimization, and deployment of thermal forward vision systems for advance vehicular applications on edge devices," 2023. [Online]. Available: https://arxiv.org/abs/2301.07613

[22] J. Zhang, T. S. White, H. Zhang, W. Hu, D. C. W. II, and J. Liu, "Ms-yolo: Infrared object detection for edge deployment via mobilenetv4 and slideloss," 2025. [Online]. Available: https://arxiv.org/abs/2509.21696

[23] S. Boddu and A. Mukherjee, "Efficient edge deployment of quantized yolov4-tiny for aerial emergency object detection on raspberry pi 5," 2025. [Online]. Available: https://arxiv.org/abs/2506.09300

[24] H. T. Dinh and E.-T. Kim, "A lightweight network based on yolov8 for improving detection performance and the speed of thermal image processing," *Electronics*, vol. 14, no. 4, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/14/4/783

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 213–229.

[26] W. Lv, S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, and Y. Liu, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 965–16 974.

[27] Z. Zong, G. Song, and Y. Liu, "Detrs with collaborative hybrid assignments training," 2023. [Online]. Available: https://arxiv.org/abs/2211.12860

[28] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," 2021. [Online]. Available: https://arxiv.org/abs/2010.04159

[29] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," 2024. [Online]. Available: https://arxiv.org/abs/2303.05499

[30] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," 2022. [Online]. Available: https://arxiv.org/abs/2203.03605

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[32] I. Robinson, P. Robicheaux, M. Popov, D. Ramanan, and N. Peri, "Rf-detr: Neural architecture search for real-time detection transformers," 2026. [Online]. Available: https://arxiv.org/abs/2511.09554

[33] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 549–565.

[34] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

[35] BIRDSAI, "Birdsai dataset," https://universe.roboflow.com/birdsai/birdsai-duqdg , jun 2022, visited on 2026-02-15. [Online]. Available: https://universe.roboflow.com/birdsai/birdsai-duqdg

[36] S. Speth, A. Alves Gonçalves, B. Rigault, S. Suzuki, M. Bouazizi, Y. Matsuo, and H. Prendinger, "Deep learning with rgb and thermal images onboard a drone for monitoring operations," *Journal of Field Robotics*, vol. 39, pp. 840–868, 05 2022.

[37] I. Kurmi, D. Schedl, and O. Bimber, "Airborne optical sectioning," *Journal of Imaging*, vol. 4, p. 102, 08 2018.

[38] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baselines," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[39] J. Morales, R. Vázquez-Martín, A. Mandow, D. Morilla-Cabello, and A. García-Cerezo, "The UMA-SAR dataset: Multimodal data collection from a ground vehicle during outdoor disaster response training exercises," *The International Journal of Robotics Research*, vol. 40, no. 6-7, pp. 835–847, 2021.

[40] Teledyne FLIR, "Free thermal dataset for algorithm training," https://oem.flir.com/solutions/automotive/dataset/, 2024, accessed: Feb. 14, 2026.

[41] J. Suo, T. Wang, X. Zhang, H. Chen, W. Zhou, and W. Shi, "Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicle-based object detection," *Scientific Data*, vol. 10, no. 1, Apr. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41597-023-02066-6