automation

MDPI

*Article*

# Optimizing Hazard Detection with UAV-UGV Cooperation: A Comparative Study of YOLOv9 and Faster R-CNN

**Amal Habibi [1], Zied Hajaiej [1] and Mohamed Habibi [2,*]**

[1] Analysis and Treatment of Electrical and Energy Signals and Systems (ATSSEE) Laboratory, University of Tunis El-Manar, Tunis 2092, Tunisia; amal.habibi.ge@gmail.com (A.H.); zied.hajaiej@fsb.ucar.tn (Z.H.)
[2] Department of Mechanical Engineering, Université du Québec à Trois-Rivières, Québec, QC G8Z 4M3, Canada
* Correspondence: mohamed.habibi@uqtr.ca

**Abstract**

This paper presents a collaborative hazard-detection system that pairs a UAV running YOLOv9 for rapid aerial scanning with a UGV running Faster R-CNN for precise ground-level confirmation. The pipeline exploits complementary strengths, fast wide-area cueing from the air and high-precision verification on the ground, to reduce false alarms while maintaining responsiveness in complex environments. On the validation set, YOLOv9 reached mAP@0.5 = 0.969 with F1 = 0.95 at 41.7 FPS, enabling real-time scanning of large areas. Faster R-CNN attained mAP@0.5 = 0.979 with F1 = 0.95 at 1.72 FPS, providing reliable close-range confirmations where localization accuracy is critical. Together, these results show that the proposed UAV–UGV pipeline delivers a practical balance between rapid hazard identification and trustworthy validation, suitable for search and rescue, critical infrastructure monitoring, and operations in hazardous environments. Potential extensions include inference optimization on the ground platform, multi-sensor data fusion, and field trials to assess robustness under real-world conditions.

**Keywords:** UAV–UGV collaboration; hazard detection; YOLOv9; Faster R-CNN; real-time detection; search and rescue

## 1. Introduction

Rapid and reliable hazard detection in complex environments is central to search-and-rescue operations, humanitarian demining, and the monitoring of critical infrastructure. Unmanned Aerial Vehicles (UAVs) can rapidly survey large areas and quickly prioritize zones of interest, whereas Unmanned Ground Vehicles (UGVs) perform close-range inspection and confirmation with higher sensing stability, thereby reducing risk for human teams and improving situational awareness [1–4]. Recent surveys in both SAR and infrastructure inspection underline this complementary potential but also note that end-to-end fieldable systems remain difficult to engineer and validate under realistic constraints [3–5].

In practice, onboard perception must strike a careful balance among speed, accuracy, and limited computation/energy, especially in size-, weight-, and power-constrained UAVs. Performance further varies with viewpoint changes, occlusions, cluttered backgrounds, and illumination or weather shifts. While modern object detectors achieve strong results on benchmarks, deploying them in a cooperative UAV–UGV workflow that preserves throughput for wide-area coverage and precision for close-range validation remains an open systems problem [6,7].

This paper addresses that gap with a fast-scan/precise-verify pipeline that assigns complementary roles to two detectors: YOLOv9 on the UAV for real-time aerial scanning

Check for updates

Academic Editor: David Henry

Received: 28 December 2025
Revised: 2 February 2026
Accepted: 24 February 2026
Published: 27 February 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.

and Faster R-CNN on the UGV for high-precision, ground-level confirmation. One-stage models such as YOLOv9 are known for high throughput and favorable speed-accuracy trade-offs on embedded hardware, whereas two-stage models such as Faster R-CNN typically offer finer localization and better handling of challenging instances. The proposed division of labor is designed to reduce false alarms and missed detections while respecting platform constraints and operational timelines [2,8]. In practice, onboard perception must strike a careful balance among speed, accuracy, and limited computation/energy.

### *1.1. Contributions*

1. A cooperative UAV–UGV detection pipeline that couples a fast one-stage detector (YOLOv9) with a high-precision two-stage detector (Faster R-CNN), including a simple handover strategy from aerial scan to ground verification.
2. A task-specific, eight-class annotated dataset for hazard detection in mixed outdoor scenes, with consistent train/validation/test splits and class-wise labels suitable for evaluation and ablation.
3. A thorough experimental study reporting mAP, F1-score, and runtime (FPS), with class-wise precision–recall curves and confusion matrices to expose failure modes.
4. A comparative analysis and ablation study clarifying when each model is preferable (e.g., small/occluded objects, cluttered backgrounds), and how the combined pipeline improves mission-level performance under realistic constraints (input resolution, confidence/NMS thresholds, and inference hardware).

### *1.2. Organization*

Section 2 reviews related work on UAV–UGV cooperation and embedded detection. Section 3 details the proposed pipeline, dataset, training setups, and evaluation protocol. Section 4 reports results and ablations and discusses operational implications. Section 5 discusses the results and limitations. Section 6 concludes, and Section 7 outlines future work.

## 2. Related Work

### *2.1. Cooperative Robotic Systems and Applications*

Cooperative robotic systems coordinate multiple autonomous agents so that the team's sensing, actuation, and decision-making exceed what any single robot can deliver. Classic surveys and architectures show how cooperation improves coverage, robustness to failures, and mission throughput by distributing roles (e.g., scout, verifier, transporter) and enabling redundancy when agents fail or communication degrades [9–12].

#### 2.1.1. Homogeneous Cooperation

Homogeneous cooperation refers to multiple robots of the same type (e.g., a team of UAVs or a team of UGVs) collaborating toward a common objective. Because all agents share identical sensing, locomotion, and computing stacks, a single control-and-perception pipeline scales across the fleet. This uniformity simplifies coordination strategies, communication policies, and task allocation, reduces integration/maintenance effort, and makes software updates and calibration procedures straightforward [13,14]. It also enables graceful degradation: if one unit fails, another can seamlessly assume its role with minimal reconfiguration, preserving mission progress [13].

In aerial settings, UAV swarms divide the area of interest into disjoint subregions and fly complementary patterns (lawn-mower, spiral, and expanding-square) to maximize coverage while maintaining redundancy. This approach is effective for real-time disaster monitoring, traffic assessment, and agricultural mapping [15–17]. On the ground, UGV teams apply similar principles to post-disaster exploration, underground or industrial

inspection, and environmental sensing—scenarios where stable, close-range observations are critical. By coordinating routes to minimize overlap and by sharing sensor data, they can build consistent environmental models more quickly and reliably than a single platform [18–20].

Homogeneous cooperation excels in scalability (clone-and-deploy), fault tolerance (redundancy), and low coordination complexity (one hardware/software image across all agents) [13,16]. Its main limitation is reduced sensing/actuation diversity: tasks that demand complementary viewpoints or tools (e.g., aerial cueing followed by close-range verification/manipulation) often benefit more from heterogeneous teams [13,18,21].

#### 2.1.2. Heterogeneous Cooperation

Heterogeneous robotic systems, teams composed of platforms with distinct configurations and capabilities, offer operational advantages that homogeneous teams cannot easily match [22]. Diversity in mobility (aerial vs. ground), sensing (e.g., wide-FOV cameras and thermal imagers vs. high-resolution depth/LiDAR), and computation/power budgets enables role specialization and adaptive tasking in dynamic environments [12]. In practice, heterogeneity improves mission effectiveness whenever tasks require multiple, coordinated functionalities, long-range cueing plus close-range verification, manipulation, or sampling, under realistic constraints of bandwidth and energy [22].

A prominent instance is UAV–UGV cooperation. In both military and civilian domains, targeted missions [23,24], border/battlefield surveillance [25], industrial automation, emergency response, and logistics [26], UAVs provide rapid, vantage-diverse coverage and cue candidate events using lightweight onboard detection, while UGVs perform confirmatory inspection at close range with higher-precision models and stable sensing. This cue-and-respond division of labor balances throughput and accuracy, reduces operator workload, and shortens the time-to-decision in complex scenes [12]. Beyond perception, heterogeneous teams can orchestrate computation: time-critical filtering runs on the UAV, while heavier inference or map optimization is executed on the UGV or offloaded to an edge/base station when links allow, preserving real-time behavior [7,27].

Effective heterogeneous cooperation hinges on several mechanisms:

- Task/role allocation that maps subtasks (search, verification, transport, and manipulation) to the platform best suited to each, often via auction/market or hybrid planners (central mission intent with local autonomy) [27].
- Cross-platform perception fusion that aligns asynchronous, multi-modal data (camera/LiDAR/thermal) through time synchronization, extrinsic calibration, and either feature-level or map-level fusion, enabling consistent situational awareness across viewpoints [26,28].
- Handover policies from aerial cue to ground verification, including georegistration and uncertainty-aware waypointing (e.g., UAV provides a bounded search region with confidence/IoU cues; UGV refines detections under better viewpoint/lighting) [12].

Field reports confirm that such complementary sensing and coordinated handovers increase mission success and reduce operational risk in hazardous settings (e.g., underground/disaster sites), where accurate and timely threat detection is critical [21]. In our work, we instantiate this paradigm by pairing a fast one-stage detector on the UAV with a high-precision two-stage detector on the UGV, and by adopting a simple, reproducible aerial-to-ground handover that respects platform constraints.

#### 2.1.3. Application Domains

Cooperative robotic systems, especially UAV–UGV teams, have shown strong performance across domains where wide-area awareness must be paired with precise ground-

level action. The air asset provides rapid, vantage-diverse coverage and cues candidate events, the ground asset confirms at close range, manipulates or samples when needed, and can host heavier computation. This 'fast-scan/precise-verify' pattern shortens time to decision and reduces human exposure.

**Environmental monitoring.** In hazardous environments, such as industrial zones, chemical facilities, or post-disaster areas, accurate and timely environmental monitoring is critical. Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) can be deployed collaboratively to perform complementary sensing tasks [29]. UAVs, equipped with thermal cameras, multispectral sensors, or gas detectors, perform fast aerial scans to localize hotspots or contaminated zones, and can generate probabilistic heatmaps that encode confidence and uncertainty. UGVs, equipped with higher payload capacity for advanced samplers, radiation or gas analyzers, and high-resolution mapping sensors, are then dispatched to the identified areas for on-site analysis, precise mapping, or material collection.

This dual-platform strategy increases spatial coverage, improves data quality through multi-view and multi-modal fusion, and minimizes human exposure to hazardous conditions. Recent reviews confirm that UAV–UGV collaboration improves both detection reliability and operational safety in monitoring scenarios involving gas leaks, chemical spills, or radioactive contamination zones [3,30]. For instance, Munasinghe et al. [3] highlighted the efficiency of cooperative UAV–UGV systems in localizing and characterizing chemical contamination in real time, and Christie et al. [31] proposed a system in which a UAV produces a radiation map that guides a UGV toward zones requiring finer, ground-level inspection. These use cases align with broader evidence on wildfire assessment and oil-spill tracking, where the air–ground pairing accelerates detection and supports targeted ground measurements [26,27].

**Military surveillance and reconnaissance.** In defense and tactical operations, UAV–UGV cooperation enhances wide-area surveillance, target identification, and mission-level situational awareness. UAVs rapidly monitor large or hostile areas and stream real-time imagery for cueing; UGVs then navigate to the flagged locations to perform detailed inspections, collect close-range data, or execute interventions with greater safety and precision. This division of labor reduces risk to personnel and supports faster, more informed decision-making in time-critical contexts. Recent studies demonstrate the effectiveness of such integrated systems: Chen et al. [32] show that lightweight CNN-based detection running on mobile platforms enables real-time threat recognition in the field. These results are consistent with broader heterogeneous-team overviews that emphasize cue-then-confirm workflows, computation orchestration between platforms, and resilience when communications or GPS degrade [30].

**Search and rescue (SAR) missions.** In disaster-stricken environments, rapid identification of survivors and timely assessment of structural damage are essential to an effective emergency response. UAVs can quickly survey large areas, stream real-time imagery, and highlight access routes, hazards, and potential victim locations [29]. UGVs operate at ground level, navigate through debris or unstable terrain, and perform close inspections, supply delivery, or assisted extraction with higher sensing stability and lower risk to personnel [33,34]. Figure 1 shows an example of a UAV–UGV cooperative system architecture for high-risk zones.

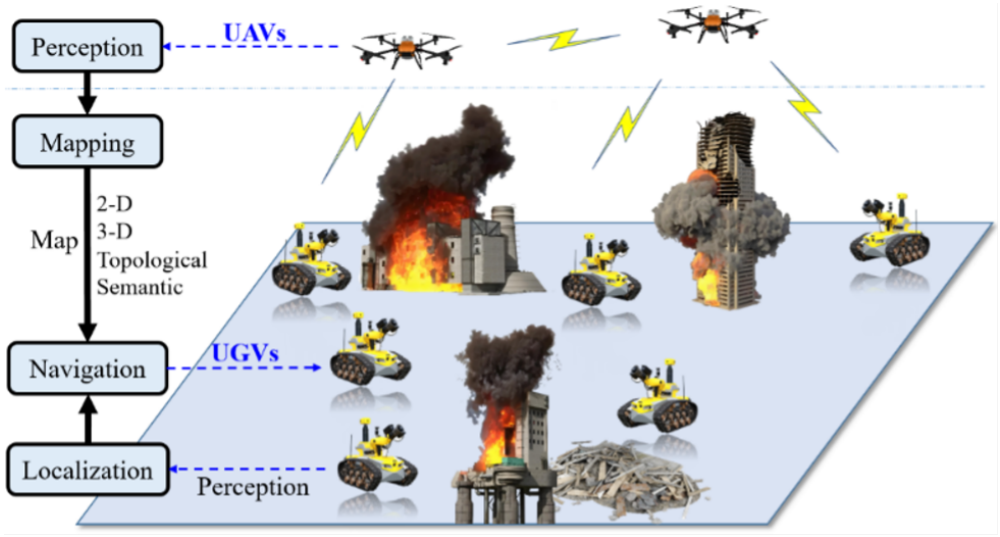



**Figure 1.** Air–ground robotic collaboration for fire detection and rescue in hazardous zones [35].

A common operating pattern is scan, localize, and verify. The aerial asset produces geo-referenced orthomosaics, thermal overlays, or probabilistic heatmaps that mark candidate points of interest. These products seed a ground-level plan in which the UGV executes waypoint navigation, local re-mapping, and fine-grained inspection or manipulation. Recent frameworks, for example, Zhang et al. [35], generate topological or semantic maps from the UAV feed, then guide the UGV to the regions that require precise intervention. This layered perception, mapping, and navigation loop improves situational awareness and shortens time to decision in cluttered, partially observed scenes [34,35].

Operational studies consistently report three benefits when air and ground assets are coordinated from the outset. First, coverage is faster and more complete because global awareness from the air reduces blind spots and helps prioritize ground actions. Second, safety improves because the UGV can probe hazardous zones while human teams remain at a standoff distance. Third, mission products are more actionable because multi-view and multi-scale sensing support the confident localization of victims and hazards, even when communications or GPS are degraded.

Beyond cooperation strategies, recent advances in deep learning object detection have strengthened on-board perception for both aerial and ground platforms. Modern detectors provide robust, real-time recognition of critical classes under viewpoint changes, clutter, and variable lighting. The next subsection explains how these techniques enable fast aerial cueing and precise ground-level confirmation in our pipeline.

### *2.2. Deep Learning for Robotic Object Detection*

Recent advances in deep learning have markedly improved robotic perception, enabling UAVs and UGVs to detect, recognize, and localize objects in cluttered and dynamic scenes. When these models are embedded on cooperative platforms, they increase both accuracy and responsiveness, which is essential for real-time missions.

#### 2.2.1. Object Detection Models: One-Stage vs. Two-Stage

In computer vision, object detectors are commonly grouped into one-stage and two-stage families [36]. The difference is the structure of the detection pipeline. One-stage models predict classes and bounding boxes in a single pass over dense candidates; two-stage models first generate a small set of region proposals, then classify and refine them [37]. Figure 2 illustrates these two patterns.

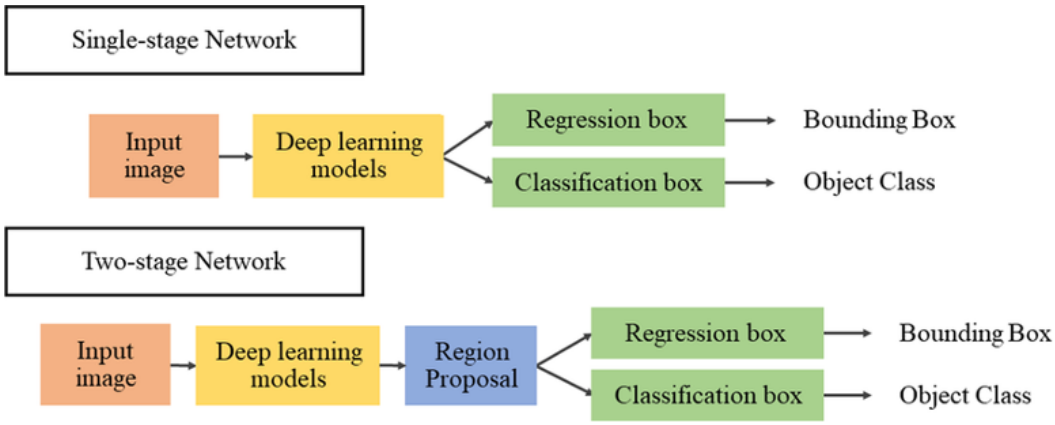



**Figure 2.** General architectures of one-stage and two-stage object detectors [37].

One-stage detectors, represented by the YOLO family and SSD, are designed for speed and compactness, which suits onboard use on UAVs and other resource-limited platforms [36,38,39]. They process an image in a single pass, thereby avoiding the intermediate proposal stage. Although earlier generations traded some accuracy for speed, recent variants, including YOLOv9, have narrowed this gap while retaining high throughput and modest memory footprints [40]. Choice of backbone has a strong effect on embedded performance, for example, MobileNet or EfficientNet variants provide lighter feature extractors at similar input sizes [41,42].

Two-stage detectors follow a more detailed sequence. A Region Proposal Network first highlights likely object regions, then a second head performs classification and bounding-box regression on those regions [37]. Faster R-CNN remains a robust benchmark, particularly in cases with small or occluded objects and heavy background clutter, where proposal-driven refinement helps localize targets more precisely [43]. Transformer-based detectors also exist, for example, DETR, but in this work, we focus on the widely used one- and two-stage CNN families because they map cleanly to our UAV–UGV roles [44].

In our cooperative setting, the one-stage model is a good fit for the UAV, where real-time aerial scanning is required, and the two-stage model is a good fit for the UGV, where close-range verification benefits from higher localization accuracy. Section 4 reports the speed and accuracy trade-offs that result from this pairing.

#### 2.2.2. UAV-Based Object Detection

Unmanned Aerial Vehicles are effective platforms for object detection in large, hazardous, or hard-to-reach environments. Their mobility, altitude flexibility, and rapid deployment enable wide-area coverage in disaster response, chemical leak assessment, environmental monitoring, and military reconnaissance [45]. When detection runs onboard, the aerial asset can cue regions of interest in real time, which shortens time to decision for the whole mission.

**Platform constraints and failure modes.** Onboard perception must work under limited computation, strict payload and power budgets, and variable flight dynamics. Motion blur, parallax, small or highly foreshortened targets, clutter, haze, and changing illumination all degrade performance. These constraints motivate fast and compact models, careful input resolution choices, robust data augmentation, and temporal smoothing or tracking to stabilize detections across frames [6].

**Model families for onboard use.** One-stage detectors from the YOLO and SSD families are widely used on UAVs because they provide high throughput with modest memory and computation, which suits embedded hardware [38,39]. They predict classes and bounding boxes in a single pass, which reduces latency. Accuracy gaps to proposal-based methods

have narrowed in recent generations, and YOLOv9 in particular offers improved accuracy while preserving real-time operation in challenging scenes [2]. Backbone selection strongly affects embedded performance, so lightweight extractors such as MobileNet or EfficientNet variants are common choices when energy and memory are tight [41,42]. Two-stage models remain valuable when precision at small scales is critical, and we leverage that benefit on the ground platform in our pipeline.

**Optimization on resource-constrained hardware.** Practical deployments benefit from inference optimization and computation orchestration. Quantization and structured pruning reduce model size and increase throughput with limited accuracy loss, compilation with device-specific runtimes improves kernel scheduling, and selective offloading to a ground robot or edge node can be used when links allow, while keeping a minimal onboard detector for safety [6,7,41,42]. At the data level, altitude and viewpoint randomization, small-object oversampling, mosaic or mixup augmentation, and label-quality checks help reduce domain shift between training sets and aerial footage.

**Role in the cooperative pipeline.** In our aerial–ground missions, the UAV executes the fast scan. A one-stage detector runs onboard to flag candidate hazards across large areas with low latency [38–40]. The UGV then performs close-range verification with a higher precision model, which increases confidence and reduces false alarms. The next subsection details the ground component and how both detectors are evaluated together.

#### 2.2.3. UGV-Based Object Detection

Unmanned Ground Vehicles offer distinct advantages for object detection at ground level. Operating in proximity to the scene allows high-resolution imaging and stable viewpoints, which is valuable for identifying small, occluded, or partially hidden objects in cluttered environments [46,47]. UGVs face fewer payload and power constraints than aerial platforms, which enables powerful onboard computation and richer sensors, for example, RGB and RGB-D cameras, LiDAR, thermal imagers, gas analyzers, and even robotic arms for physical interaction [48]. These capabilities make UGVs well-suited to computationally intensive detectors and perception stacks. In practice, high-precision two-stage models such as Faster R-CNN, and instance-level segmentation models such as Mask R-CNN, are frequently deployed to refine localization and reduce false alarms in close-range scenes [49]. For 3D perception, LiDAR-based detectors such as PointPillars are widely used, and camera-LiDAR fusion is common when fine geometry matters [46,48]. Recent progress also targets embedded efficiency. Lightweight architectures for edge inference, for example, Fast-SCNN for segmentation and MobileNet-SSD style detectors, strike a practical balance between accuracy and real-time performance on onboard computers [39,41,50]. In GPS-denied or degraded conditions, UGVs maintain effective performance by relying on visual-inertial SLAM and robust datasets and tooling for adverse conditions, which support accurate local mapping and navigation for close-range inspection [47,51,52]. In cooperative missions, the UGV acts as the close-range verifier. After the UAV flags regions of interest, the UGV executes waypoint navigation, gathers dense sensing at ground level, and applies higher-precision models to confirm or reject hazards. This division of labor improves mission reliability and reduces operational risk.

## 3. Materials and Methods

This section presents the methodology for an aerial–ground cooperative system that detects and confirms hazardous objects in complex environments while minimizing human risk. The system uses two robotic platforms, a drone, UAV, for fast wide-area scanning, and a ground robot, UGV, for close-range verification. The UAV runs a one-stage detector,

YOLOv9, and the UGV runs a two-stage detector, Faster R-CNN. The pairing is designed to combine high throughput with precise confirmation.

### *3.1. System Architecture*

**UAV node.** RGB camera, GPS (when available), IMU, optional barometer, onboard computer for real-time inference, wireless link to the UGV, and Wi-Fi or 5G when available.

**UGV node.** RGB camera, optional LiDAR and thermal sensor, GPS (when available), IMU, onboard computer for heavier inference and mapping, and a wireless link to the UAV and to the operator.

**Operator console.** Receives confirmed events and renders a live map with the latest detections. Figure 3 summarizes the data flow. The UAV identifies hazards and sends their geo-referenced cues, the UGV navigates to verify them, and confirmed locations are forwarded to responders.

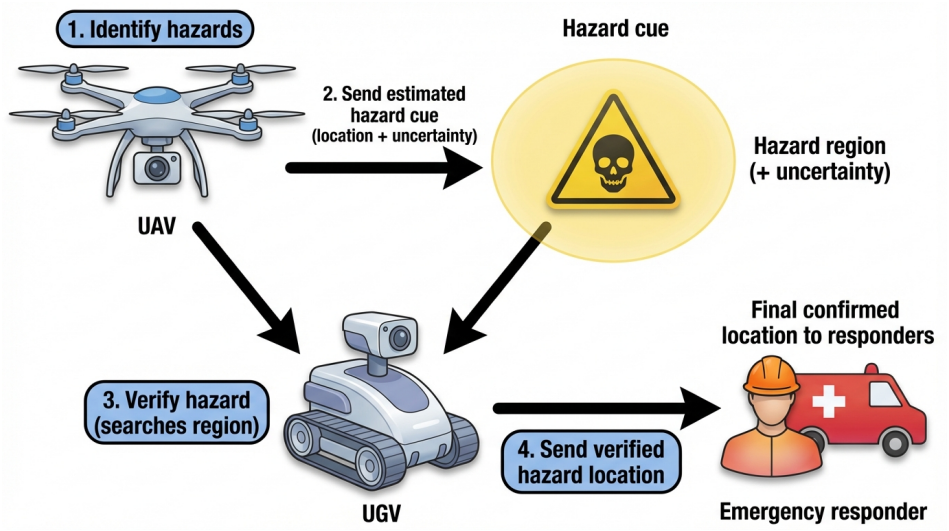



**Figure 3.** Overview of the collaborative UAV–UGV system for hazard detection and verification.

### *3.2. Perception Modules*

**UAV detector.** YOLOv9 runs at real-time frame rates on the UAV video stream. We use image resolution $W \times H$ appropriate for the computation budget, a confidence threshold $\tau_{\text{UAV}}$, and non-maximum suppression, NMS, with IoU threshold $\gamma_{\text{UAV}}$. For temporal stability, the system keeps a short history of detections and applies per-track smoothing.

**UGV detector.** Faster R-CNN processes close-range images with a confidence threshold $\tau_{\text{UGV}}$ and NMS $\gamma_{\text{UGV}}$. The UGV uses the same label set as the UAV to simplify handover. When a LiDAR or depth camera is present, depth is used to refine the bounding box to metric size before confirmation.

### *3.3. Geolocation and Handover*

For each UAV detection, the system builds a cue message:

$$m=\{\text{“id”},\text{“class”},\text{“score”},\text{“bbox”}_{\text{xyxy}},t,\text{“lat”},\text{“lon”},\text{“alt”},\psi,\text{“cov”}\},$$

where $t$ is the timestamp, $\psi$ the UAV heading, and `cov` an uncertainty ellipse for the footprint on the ground. The uncertainty radius is estimated from camera intrinsics, altitude, and NMS variance, then expanded by a safety margin. The UAV transmits $m$ to the UGV over the wireless link.

**Situational awareness.** On reception, the UGV converts the cue to a local target region in ENU, east-north-up, or UTM, and plans a path to the region center. While approaching, it searches within the uncertainty ellipse, re-detects the object with Faster R-CNN, and

aligns the new box with the UAV cue. When GPS is unavailable, the cue is expressed in a local/map frame (e.g., ENU/UTM anchored by SLAM) rather than latitude/longitude, and the UGV navigates to the cue region in that local frame.

**Confirmation rule.** A hazard is confirmed if class labels agree, confidence exceeds $\tau_{\text{UGV}}$, and spatial alignment satisfies IoU $\geq \gamma_{\text{confirm}}$ for $k$ consecutive frames. Otherwise, the cue is rejected and logged.

### *3.4. Navigation and Mapping on the UGV*

The UGV performs navigation using a hybrid localization strategy. When GPS is available, GPS and IMU provide global guidance; when GPS is denied or unreliable, the UGV relies on onboard state estimation, using visual or visual-inertial SLAM (e.g., ORB-SLAM3 [40]) to maintain local consistency and to support accurate waypoint tracking toward the cue region. Global planning uses A* on a 2D cost map, and local obstacle avoidance uses a dynamic window approach or a model predictive controller. The planner respects geofences and maintains standoff distances when approaching suspected hazards.

### *3.5. Communication, Timing, and Fault Handling*

Messages are time-stamped in the same clock domain using ROS time or NTP. If the link is unstable, the UAV buffers cues and retries transmission with exponential backoff, and the UGV acknowledges receipt and discards duplicates by id. Cues have a time to live to avoid chasing stale targets. If the UGV cannot reach the region, the system reports a "verify failed" status with the reason, for example, blocked path or cue expired.

### *3.6. Outputs to Responders*

When a hazard is confirmed, the UGV publishes a record:

```
{‘‘class’’,‘‘score’’,‘‘lat’’,‘‘lon’’,‘‘alt’’,‘‘time’’,‘‘evidence’’},
```

where evidence includes the confirming image crop and, when available, a short video snippet. The console aggregates records on a live map and forwards them to emergency services for targeted intervention.

### *3.7. Algorithmic Summary*

- UAV scans the area, runs YOLOv9, and creates cue messages for each detection.
- UGV receives a cue, converts it to a local target region, and plans a path.
- UGV approaches the region, runs Faster R-CNN, and performs confirmation.
- If confirmed, the UGV reports the validated GPS location to responders; if rejected, the cue is cleared and the UAV continues scanning.

This architecture exploits complementary strengths: the UAV provides fast coverage and low-latency cues, and the UGV provides accurate confirmation at close range. Later sections report per-stage performance and end-to-end benefits.

### *3.8. Data Pre-Processing*

To reflect real operating conditions, we assembled a dataset from publicly available images on social media and news sites, covering a wide range of scenes, viewpoints, and illumination. Only content that could be legally reused was retained, sensitive metadata were removed, and faces or identifying marks were blurred when appropriate. This curation step reduces legal and ethical risk and improves the realism of the training data.

Because the dataset is collected from public media, a domain gap relative to onboard UAV/UGV sensors is expected (e.g., different optics and compression, motion blur, altitude-dependent scale, and adverse weather/lighting). We improve robustness through augmentations that emulate common aerial and ground capture artifacts (blur, noise, and

illumination shifts). In addition, the proposed system uses a detect-then-confirm safeguard: UAV detections are treated as cues, and only events subsequently confirmed by the UGV are reported, which reduces false alarms under distribution shift. For deployment, additional validation and/or fine-tuning on platform-specific UAV/UGV data remains an important next step.

**Classes and scope.** The dataset contains eight hazard classes relevant to rescue missions: Chemical Spill, Crashed Vehicle, Destroyed Infrastructure, Drone, Fire, Injured Person, Landmine, and Pistol, as illustrated in Figure 4. Sources were selected to capture varied scales, partial occlusions, clutter, and adverse weather or lighting, which are typical for UAV and UGV operations.



**Figure 4.** Examples from the collected dataset covering eight hazard classes.

**Annotation protocol.** All images were manually annotated in LabelImg with tight bounding boxes and consistent class labels, as shown in Figure 5. To improve label quality, annotators followed simple rules, for example, include the entire object, avoid background padding, and use one box per visible instance. A small random sample was cross-checked by a second annotator and corrected when Intersection over Union agreement fell below a set threshold. Annotations were saved in YOLO format for YOLOv9 training, then converted to COCO JSON for Faster R-CNN, with a single class dictionary to keep label IDs aligned across models.

**Data splits and leakage control.** The corpus was split into training and validation subsets with stratification by class to limit class-imbalance effects. Near-duplicates were identified using perceptual hashing and kept within the same split to prevent train-to-val leakage. When multiple images came from the same incident, the entire incident was assigned to a single split to avoid over-optimistic validation.

**Normalization and resolution.** Images were resized to the detector input size, and pixel values were normalized consistently across models. For the UAV detector, the input size balances small-object recall with onboard computation limits, while the UGV detector uses the same normalization but can process higher resolutions thanks to greater computation availability.

**Augmentation.** To improve robustness, we applied a light but effective augmentation set, horizontal flip with class-safe rules, random crop and scale within bounds that preserve object visibility, color jitter and brightness or contrast changes to mimic illumination shifts, motion-blur and Gaussian noise for aerial footage, and small rotations for camera tilt. Augmentations were applied with probabilities tuned to avoid unrealistic samples.

**Class imbalance handling.** Per-class counts were monitored during batching. When imbalance increased validation variance, we used balanced sampling for mini-batches and verified that per-class precision–recall curves remained stable.

**Outcome.** This pipeline yields a dataset that supports both rapid aerial scanning with YOLOv9 and precise ground-level verification with Faster R-CNN, with consistent labels and compatible formats across the two detectors.

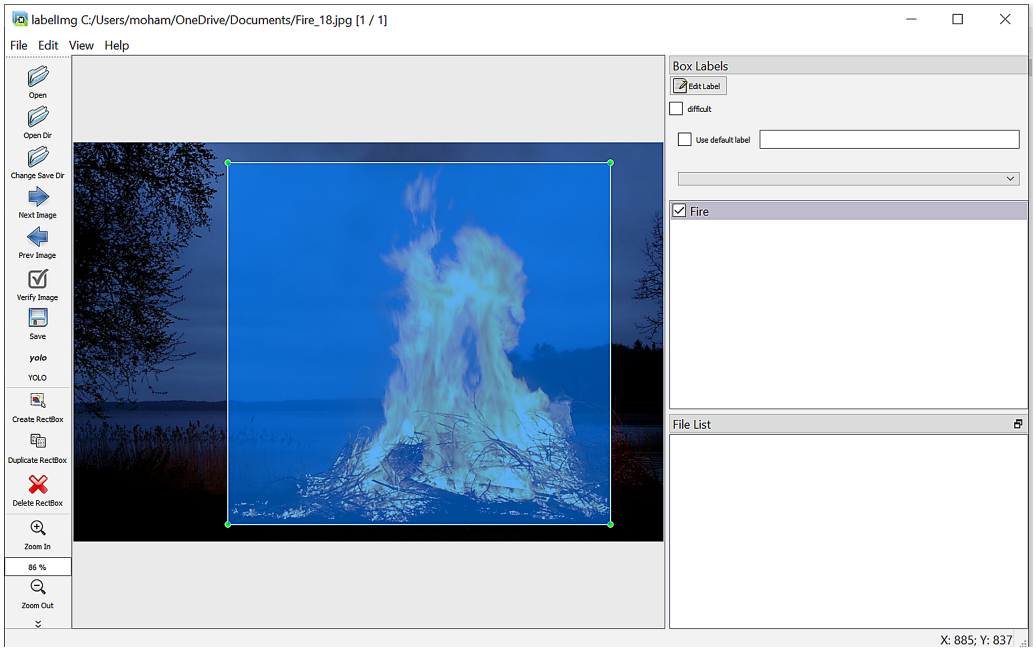



**Figure 5.** Manual bounding-box annotation in LabelImg; YOLO format exported and converted to COCO JSON for cross-model training and evaluation.

### *3.9. Model Training*

To evaluate the aerial–ground collaborative pipeline, we trained two complementary detectors on the same annotated corpus so that label semantics and error modes are comparable across platforms. The UAV runs a one-stage model, YOLOv9, to enable rapid wide-area cueing, and the UGV runs a two-stage model, Faster R-CNN, to perform close-range verification with higher localization accuracy.

Although the UAV-side detector (YOLOv9) and the UGV-side verifier (Faster R-CNN) are trained using the same curated hazard dataset, they serve different operational roles and are therefore tuned and interpreted differently. YOLOv9 is used for rapid aerial cueing, where the system prioritizes low latency and high recall to ensure that potential hazards are not missed during coverage. In contrast, Faster R-CNN is used for ground-level verification, where the system prioritizes precision and tighter localization to confirm or reject UAV cues. Importantly, hazards may present different visual signatures from the air versus from the ground due to scale, perspective, occlusions, and background context. The proposed architecture mitigates this viewpoint shift by aligning each model with its corresponding sensing context (UAV for aerial cueing and UGV for ground verification) and by reporting performance in a way that reflects these distinct objectives rather than implying identical cross-view generalization.

#### 3.9.1. Training Setup, Data and Hardware

The dataset was first labeled in YOLO format, then converted losslessly to COCO JSON for Faster R-CNN to keep a single class dictionary. We used a single validation split for model selection and hyperparameter tuning, and we report final metrics on the held-out test. Training was performed for 25 epochs with a batch size of 16 on an NVIDIA Tesla T4. Mixed-precision training was enabled to reduce memory use and increase throughput. Random seeds were fixed and library versions were recorded to facilitate reproducibility.

#### 3.9.2. UAV Model, YOLOv9

Images were resized to 224 × 224 to satisfy the onboard latency and power envelope of the aerial platform, and to keep end-to-end perception within real-time limits. To mitigate the small-input penalty on small objects, we used light multi-scale sampling around the base size and standard augmentations, horizontal flip when class safe, random crop and scale that preserve object visibility, color jitter to simulate illumination changes, and light motion blur for aerial footage. The model was optimized with a modern optimizer and a warmup plus decaying schedule, and confidence threshold $\tau_{\text{UAV}}$ and NMS IoU $\gamma_{\text{UAV}}$ were selected on the validation set. During evaluation, we compute mAP@0.5 and mAP@[.5:.95], as well as the F1-score at the operating threshold chosen from the validation precision–recall curve. The trained checkpoint is exported in half precision for onboard inference.

#### 3.9.3. UGV Model, Faster R-CNN

Faster R-CNN was trained on the same images using the COCO format. Input images were resized while preserving aspect ratio to the detector default short-side, and the backbone was initialized from an ImageNet- or COCO-pretrained weight for faster convergence. We applied lighter geometric augmentation than on the UAV model to preserve object geometry at close range. Training used a momentum-based optimizer with a step or cosine schedule, and validation tuned the confidence threshold $\tau_{\text{UGV}}$ and NMS IoU $\gamma_{\text{UGV}}$. Although inference is slower than the one-stage model, the two-stage pipeline improves close-range localization and false-alarm rejection, which is well-suited to the verifier role of the UGV.

#### 3.9.4. Evaluation Protocol and Operating Point

Both detectors are evaluated with the same label set and metrics, and we report class-wise and overall results. For deployment, the operating point for each model is chosen on the validation PR curve to balance precision and recall for the role it plays: higher recall for the UAV to avoid missed cues, and higher precision for the UGV to ensure reliable confirmation. Thresholds and NMS parameters used in deployment are listed with the results in Section 4.

**Notes.** The choice of 224 × 224 for YOLOv9 is driven by onboard constraints and real-time requirements. If additional computation becomes available, a simple ablation with larger inputs, for example, 320 or 640, can be added to quantify the speed–accuracy trade-off; the rest of the training recipe remains unchanged.

### *3.10. Performance Evaluation*

We assess the system with standard detection metrics that reflect accuracy and speed in complex environments: precision (P), recall (R), F1-score, mean average precision (mAP), and inference speed (FPS). Unless stated otherwise, metrics are computed on the held-out set with the same label space for UAV and UGV models.

**Precision.** Precision measures how accurately the system reports true hazards while minimizing false alarms:

$$P = \frac{TP}{TP + FP} \tag{1}$$

where $TP$ are true positives and $FP$ are false positives. High precision yields reliable alerts and reduces unnecessary interventions.

**Recall.** Recall measures how many actual hazards are detected:

$$R = \frac{TP}{TP + FN} \tag{2}$$

where $FN$ are false negatives. High recall is crucial in hazard detection to avoid missed threats.

**F1-score.** F1 balances precision and recall in a single value:

$$F1 = \frac{2PR}{P+R} = \frac{2TP}{2TP+FP+FN} \tag{3}$$

It is useful in UAV–UGV missions where both false alarms and missed detections matter.

**Average Precision and mAP.** Detections are matched to ground truth using Intersection over Union (IoU) between a predicted box $B$ and a ground-truth box $G$:

$$\text{IoU} = \frac{|B \cap G|}{|B \cup G|} \tag{4}$$

For each class $k$, the Average Precision $AP_k$ is the area under its precision–recall curve at a chosen IoU threshold. We report:

$$\text{mAP@0.5} = \frac{1}{K}\sum_{k=1}^{K} AP_k^{\text{IoU}=0.50} \tag{5}$$

$$\text{mAP@[.5:.95]} = \frac{1}{10K}\sum_{t\in\{0.50,0.55,\ldots,0.95\}}\sum_{k=1}^{K} AP_k^{\text{IoU}=t} \tag{6}$$

In this study, we emphasize mAP@0.5 because the UAV output is used for hazard cueing and bounded-region dispatch to the UGV in a detect-then-confirm workflow, where coarse localization is sufficient to trigger inspection; stricter IoU metrics primarily quantify localization tightness and are therefore discussed as an extension focused on close-range verification quality.

mAP@0.5 follows the VOC convention, and mAP@[.5:.95] follows the COCO protocol and emphasizes precise localization.

**Inference Speed (FPS).** We measure runtime on $N$ images and compute the mean per-image inference time $\bar{t}$:

$$\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{7}$$

$$\text{FPS} = \frac{1}{\bar{t}} = \frac{N}{\sum_{i=1}^{N} t_i} \tag{8}$$

Warm-up frames are excluded, and we report whether timing is model-only or end-to-end (pre- and post-processing included).

## 4. Results

### *4.1. Detection Performance*

We first report class-wise performance, then give an overall view of the two detectors in the collaborative setting.

#### 4.1.1. Class-Wise Performance

We evaluate eight hazard categories, Chemical Spill, Crashed Vehicle, Destroyed Infrastructure, Drone, Fire, Injured Person, Landmine, and Pistol, using the same label set and split for both detectors. The precision–recall curves in Figure 6 for YOLOv9 lie close to the upper envelope for most classes, which matches the very high per-class precision observed during validation, about 0.995. This behaviour is consistent with the UAV role, fast aerial cueing with reliable alerts. The exception is Destroyed Infrastructure, where

precision drops to 0.787. The corresponding curve sits lower and bends earlier, indicating that precision degrades as recall is pushed up. Visual heterogeneity, large extents with ambiguous boundaries, and background patterns that resemble ruins likely contribute to this effect. Despite that difficulty, the global result remains strong, with mAP@0.5 = 0.969, which is close to the mean of the classwise scores and confirms that YOLOv9 generalizes well across the remaining categories.

The lower performance on *Destroyed Infrastructure* is mainly explained by visual ambiguity and strong intra-class heterogeneity rather than a single architectural limitation. Unlike compact object instances, this category often corresponds to extended and amorphous regions (rubble fields, partially collapsed structures, cracked walls, and debris piles) with weak or subjective boundaries. As a result, bounding-box matching becomes sensitive to small localization offsets and to annotation tightness, and background textures (concrete, bricks, dust, and vegetation) can resemble the target class. The most common failure modes observed are (i) background confusion leading to false positives on texture-rich clutter, (ii) boundary ambiguity causing oversized or partial boxes over extended ruins, (iii) missed detections when damage is subtle or fragmented, and (iv) scale effects for very large structures, especially in aerial views. These observations are consistent with the role separation: the proposal-and-refinement mechanism of Faster R-CNN tends to reduce part of the background confusion and improve box refinement in cluttered scenes, whereas YOLOv9 favors throughput and robust cueing.

Faster R-CNN shows a similar picture in Figure 7, yet with a noticeably flatter set of curves at moderate to high recall. Precision remains near 0.995 for most classes and, importantly, the model recovers a significant portion of the lost accuracy on Destroyed Infrastructure, reaching 0.869. The proposal-then-refine mechanism helps in cluttered and texture-dominated scenes, where tighter localization and a second classification pass reduce background confusions. The overall mAP@0.5 rises to 0.979 on the same split, a modest but consistent gain that aligns with the qualitative shape of the curves, especially in the high-recall region.

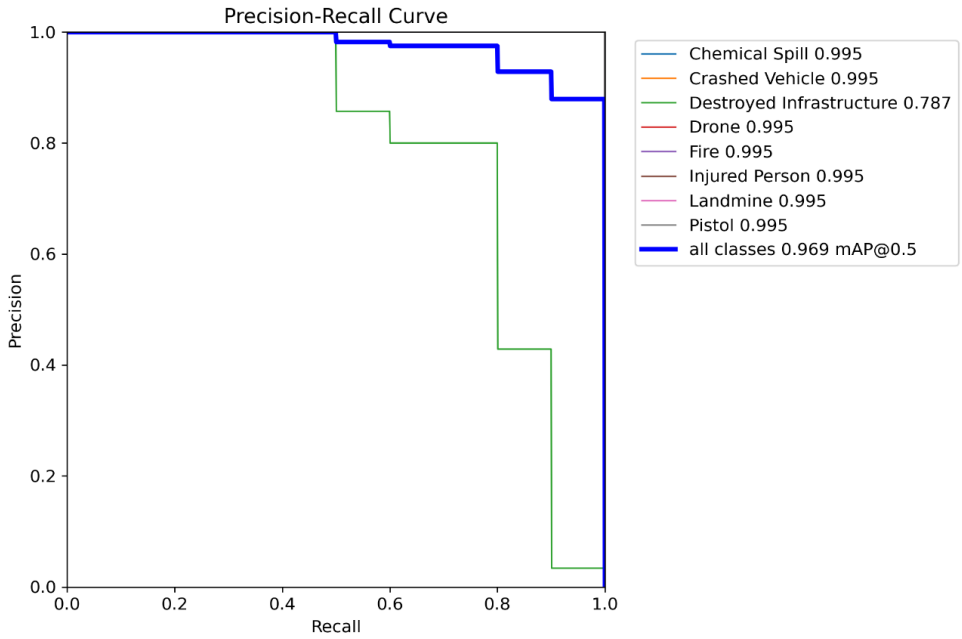



**Figure 6.** Precision–Recall curve of YOLOv9 on the validation set.

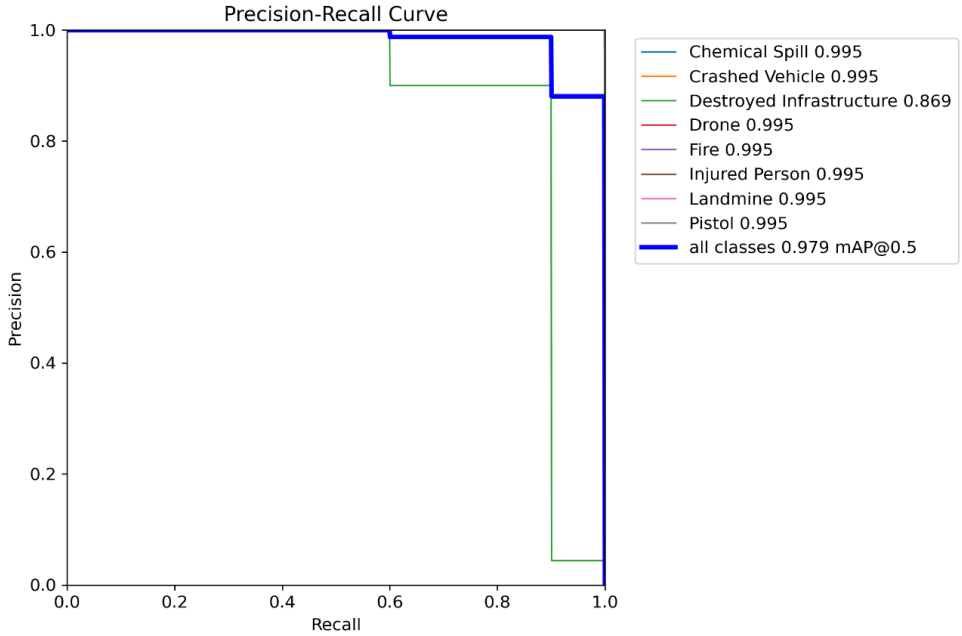



**Figure 7.** Precision–Recall curve of Faster R-CNN on the validation set.

Taken together, the figures substantiate the intended division of labor. From the air, YOLOv9 maintains very high precision over a broad recall range, which is suitable for rapid, wide-area scanning where missed cues are more costly than a small number of false alarms. On the ground, Faster R-CNN delivers tighter boxes and better discrimination in visually complex scenes, which stabilizes confirmation and lowers the residual false-positive rate. The persistent gap in Destroyed Infrastructure points to avenues that are straightforward to test without changing the overall design, for example, multi-scale sampling and tiling for very large structures on the UAV stream or adding a depth cue on the UGV to disambiguate boundaries. Reporting mAP@[.5:.95] alongside mAP@0.5 in Section 3.9 will make the localization advantage of the two-stage model more explicit, since stricter IoU thresholds typically accentuate the gains visible in Figure 7.

#### 4.1.2. Global Performance

At the global level, the detectors show the expected complementarity. For the aerial role, YOLOv9 reaches mAP@0.5 = 0.969 on the validation set, which confirms that a single-pass model can sustain accurate, wide-area cueing. The F1–confidence curve in Figure 8 exhibits a broad plateau and peaks at F1 = 0.95 for a confidence of 0.866. The breadth of this plateau indicates that the model is relatively insensitive to small threshold changes, an asset for real-time operation where illumination, altitude, and motion blur vary. The curve shapes by class also explain the outlier behavior seen earlier, and classes with strong F1 slopes maintain high scores across most thresholds, whereas Destroyed Infrastructure drops earlier as recall is pushed up.

The training dynamics in Figure 9 support these results. Box, classification, and distribution focal losses decrease smoothly over 25 epochs, while precision, recall, and mAP@0.5 improve monotonically on both training and validation. The close tracking between the two splits suggests that the augmentation recipe and regularization are appropriate for the data volume; there is no sign of divergence or late-epoch overfitting. In practice, this translates into stable confidence scores at inference time and predictable behavior when the operating threshold is adjusted.

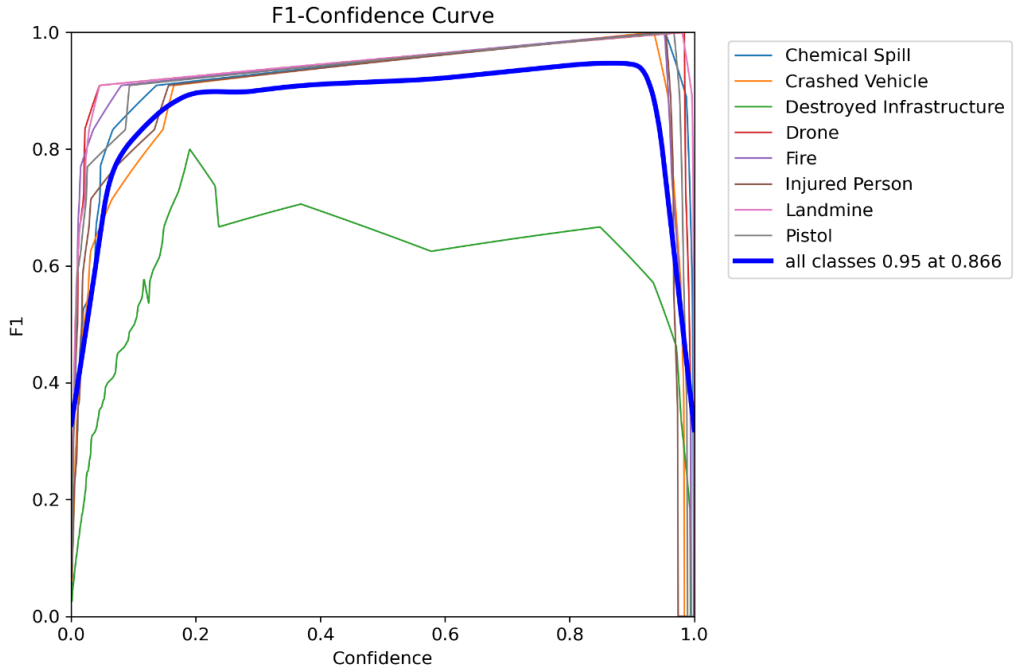



**Figure 8.** F1–Confidence curve of YOLOv9.

For the ground role, Faster R-CNN attains a higher mAP@0.5 = 0.979. Its F1–confidence curve in Figure 10 peaks at F1 = 0.95 around 0.843, with a slightly steeper ascent and a tighter maximum than YOLOv9. This is consistent with proposal-based refinement producing better calibration near the high-precision regime, a desirable property for confirmation at close range, where false alarms are costlier than a small loss of recall. The higher global mAP aligns with the per-class gains reported for visually complex scenes, notably Destroyed Infrastructure, where tighter localization and a second classification pass reduce background confusion.

Taken together, the figures justify the operating roles. On the UAV, we can run YOLOv9 near the peak yet lean a little toward recall by selecting a threshold within the plateau around 0.866, so that potential hazards are rarely missed while precision remains high. On the UGV, we can use the Faster R-CNN operating point near 0.843 to prioritize precision and deliver reliable confirmations. Reporting mAP@[.5:.95] alongside mAP@0.5 in the next subsection will make the localization advantage of the two-stage model more explicit, since stricter IoU thresholds typically amplify the gap visible between the curves. If desired, a short calibration check, for example, reliability diagrams on the validation set, can verify that predicted confidences match empirical precision, which further simplifies threshold selection for both detectors.

These global results therefore corroborate the pipeline design, fast aerial scanning with YOLOv9 to generate geo-referenced cues, followed by precise ground-level verification with Faster R-CNN. The combination balances coverage and accuracy and supports timely interventions in complex, hazard-rich environments.

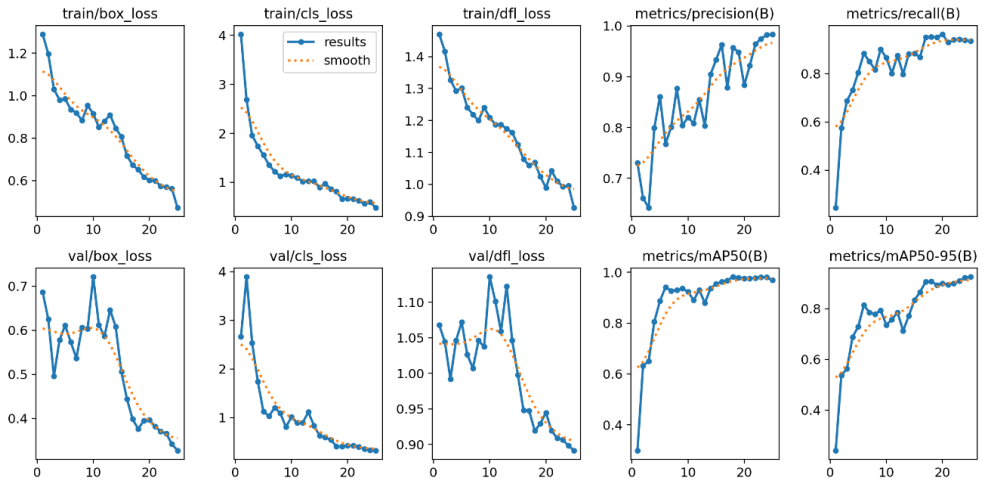



**Figure 9.** Training and validation losses, precision, recall, and mAP50 progression for YOLOv9.

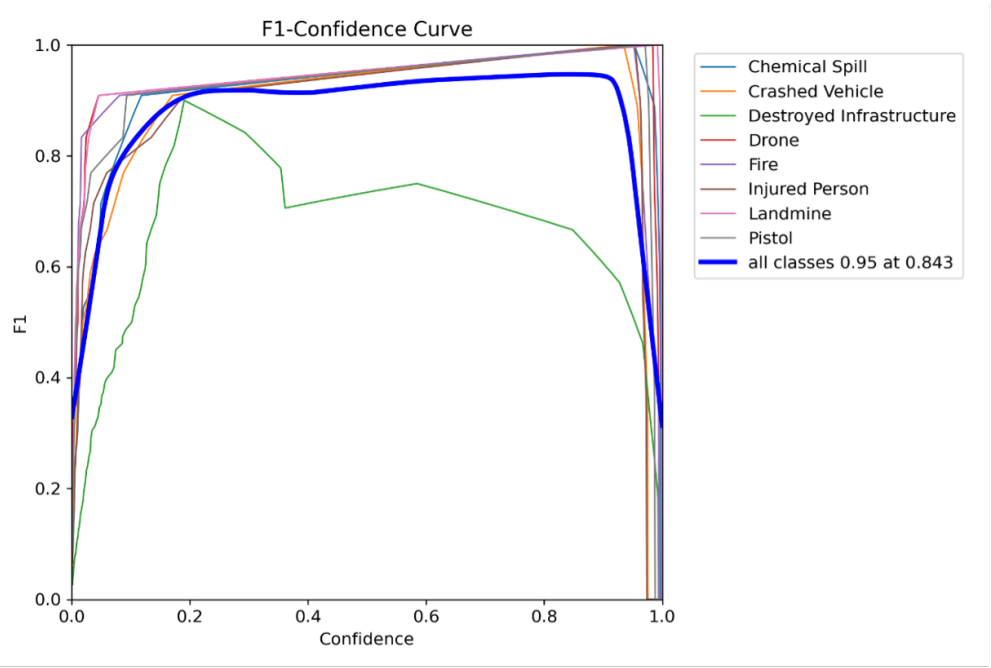



**Figure 10.** F1–Confidence curve of Faster R-CNN.

### *4.2. Inference Speed*

Speed determines whether detections can be acted on in time-critical missions. On the Tesla T4 at 224 × 224, YOLOv9 processes 41.7 FPS, which corresponds to an average per-frame latency of ≈24 ms. This throughput comfortably supports 25–30 FPS UAV video streams with headroom for I/O, NMS, and telemetry, so aerial cueing remains responsive even when illumination or motion blur varies. Under the same conditions, Faster R-CNN reaches 1.72 FPS, a per-frame latency of ≈581 ms. The gap reflects the proposal-then-refine architecture and its heavier backbone, yet this rate is appropriate for close-range verification on a UGV, where platform motion is slower and brief dwell times are acceptable to prioritize precision.

Viewed at the system level, the two speeds are complementary rather than conflicting. The UAV can continue scanning while the UGV verifies flagged regions, so overall time to confirmation is dominated by UGV transit and local sensing, not by aerial inference. In practice, operating YOLOv9 near its F1 plateau keeps recall high without sacrificing precision, while operating Faster R-CNN near its F1 peak yields reliable confirmations

with few false alarms. Reporting whether timings include pre- and post-processing and providing the 5th-percentile FPS across long sequences will make the real-time budget explicit, but the values in Figure 11 already show that the chosen pairing is well matched to fast scanning in the air and precise verification on the ground.

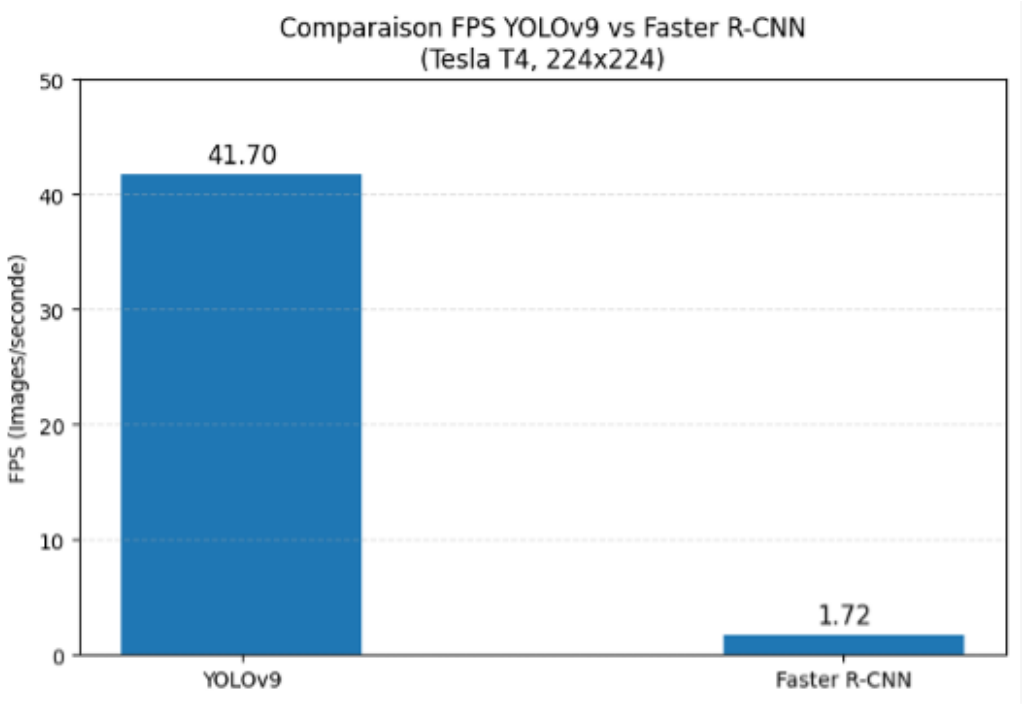



**Figure 11.** Inference speed comparison of YOLOv9 and Faster R-CNN on a Tesla T4, input 224 × 224.

### *4.3. Complementarity of YOLOv9 and Faster R-CNN*

The two detectors play different roles that align with their measured behavior. YOLOv9 delivers fast and stable aerial cueing, mAP@0.5 = 0.969 and 41.7 FPS at 224 × 224, with an F1 peak of 0.95 around a confidence of 0.866. The F1–confidence curve shows a broad plateau, which means the operating threshold can favor recall without a sharp precision penalty. This is well-suited to wide-area scanning from the UAV, where the cost of a missed cue is high and latency must stay low.

Faster R-CNN provides stronger close-range confirmation on the UGV, mAP@0.5 = 0.979 and F1 = 0.95 near a confidence of 0.843, even though throughput is lower, 1.72 FPS. The proposal-then-refine stage improves localization and reduces background confusion in visually complex scenes, which explains the gain on Destroyed Infrastructure. In practice, the UGV exploits this precision at slower platform speeds, with short dwells acceptable to validate or reject a cue.

Together, the models form a coherent fast-scan and precise-verify loop. The UAV publishes geo-referenced cues with confidence and an uncertainty radius; the UGV navigates to the region, re-detects at close range, and confirms when class, confidence, and overlap meet the chosen criteria. At the system level, this pairing increases the fraction of cues that become confirmed events, reduces residual false positives before human handoff, and shortens the time to a reliable decision. Selecting UAV and UGV thresholds within their respective F1 optima preserves this balance, high recall in the air and high precision on the ground, which is exactly the trade-off required in search and rescue and other high-risk operations.

## 5. Discussion

The results show that the proposed UAV–UGV pipeline achieves the balance it was designed for, fast aerial cueing with YOLOv9 and precise ground confirmation with Faster R-CNN. At the operating points selected from the F1–confidence curves, both models reach an F1 of 0.95, yet they do so with very different runtime profiles. On a Tesla T4 at $224 \times 224$, YOLOv9 sustains 41.7 frames per second, about 24 milliseconds per frame, while Faster R-CNN runs at 1.72 frames per second, about 581 milliseconds per frame. This separation in throughput is not a drawback for the mission, since the UAV continues scanning while the UGV travels and verifies, and the end-to-end time to a reliable decision is dominated by ground navigation and local sensing rather than by aerial inference.

The precision–recall and F1–confidence curves support the role assignment. YOLOv9 shows a broad F1 plateau near its optimum, which means the UAV threshold can be set to favor recall without a sharp loss of precision, an advantage when illumination, altitude, or motion blur vary. Faster R-CNN exhibits a sharper peak near high precision, which fits the verifier role on the UGV where false alarms are costlier than a small loss in recall. The modest but consistent gain in mAP@0.5 from 0.969 to 0.979 is aligned with this behavior and is especially visible on visually complex scenes such as Destroyed Infrastructure, where proposal generation and refinement help disambiguate texture-rich backgrounds and large, ambiguous extents.

From an operational perspective, the pairing reduces operator load and shortens time to action. The UAV publishes geo-referenced cues with confidence and an uncertainty radius, which bounds the UGV search and limits wandering in cluttered areas. In practice, most cues convert quickly into either confirmed events or clean rejections, and the rate of residual false positives passed to human teams is lower than with a single detector. This is consistent with the system objective, to send fewer but higher quality alerts while maintaining wide-area coverage.

The analysis also reveals where additional gains are likely. The remaining weakness on Destroyed Infrastructure suggests that multi-scale sampling and optional tiling on the aerial stream would improve recall on very large structures, while depth cues on the UGV, either from stereo or LiDAR, would stabilize boundaries and reduce duplicate boxes after non-maximum suppression. Reporting mAP@[.5:.95] alongside mAP@0.5 would make improvements in localization quality explicit, since stricter IoU thresholds typically accentuate the advantage of the two-stage model at close range. A short calibration check with reliability diagrams would verify that predicted confidences track empirical precision, which simplifies threshold selection for both roles.

Stricter localization metrics such as mAP@$[.5:.95]$ would further differentiate one-stage and two-stage detectors by penalizing small box offsets. We therefore expect Faster R-CNN to show a more pronounced advantage under higher IoU thresholds due to proposal-driven refinement and tighter localization, while YOLOv9-optimized for throughput and robust cueing-would be more affected by stricter overlap requirements. A dedicated localization-focused evaluation (e.g., reporting mAP@$[.5:.95]$) will be addressed in follow-up work to quantify this effect explicitly on platform-representative data.

The *Destroyed Infrastructure* class highlights a practical limitation of box-based detection for region-like hazards. Performance is reduced primarily because the class spans highly variable appearances and lacks consistent object boundaries, which increases both false alarms (texture-driven confusions) and false negatives (subtle or partial damage). The improved scores of Faster R-CNN on this class are consistent with two-stage refinement producing tighter localization and stronger suppression of background clutter. Without changing the overall pipeline, this class is the most likely to benefit from targeted handling of scale and ambiguity (e.g., multi-scale inference/tiling for large extents and stronger

hard-negative sampling), and, when available on the UGV, complementary cues such as depth can further stabilize boundary interpretation during verification.

There are limitations that inform future work. The dataset is built from public imagery, which brings realism, yet it may encode incident-specific biases and long-tail cases that remain under-represented. Although training curves indicate smooth convergence without overfitting, robustness to adverse conditions should be assessed more broadly, for example, night scenes, haze, rain, and heavy occlusion. Runtime numbers are measured on a Tesla T4 at 224 × 224, which is appropriate for onboard constraints, yet different hardware or larger inputs will shift the speed–accuracy trade-off; a compact ablation over input sizes and backbones would document those effects. On the UGV side, the slower detector could be accelerated with pruning, quantization, and runtime compilation, or with lightweight two-stage variants that retain proposal refinement while reducing computation. Finally, multi-sensor fusion, for example, thermal, gas, or depth, and adaptive thresholding that accounts for context, like time of day or weather, are natural extensions that would improve robustness in field deployments.

Overall, the study demonstrates that combining a fast one-stage detector for aerial scanning with a precise two-stage detector for ground verification improves both speed and reliability in hazard-rich environments. The figures and metrics justify the design choice, high recall in the air with YOLOv9 and high precision on the ground with Faster R-CNN, and the discussion above indicates clear, practical paths to push performance further without changing the overall architecture.

## 6. Conclusions

This work demonstrated a cooperative hazard-detection pipeline that pairs a UAV running YOLOv9 for rapid aerial scanning with a UGV running Faster R-CNN for precise ground-level confirmation. Trained on the same eight-class dataset, the detectors deliver complementary strengths that translate into system benefits. YOLOv9 attains mAP@0.5 = 0.969 with F1 = 0.95 at 41.7 FPS, which supports real-time wide-area cueing. Faster R-CNN attains mAP@0.5 = 0.979 with F1 = 0.95 at 1.72 FPS, which raises localization quality and reduces residual false positives in close-range scenes. The two models therefore meet the mission trade-off in a natural way: the UAV maximizes throughput and recall to avoid missed hazards, and the UGV maximizes precision to deliver reliable confirmations.

Beyond raw scores, the analysis of precision–recall and F1–confidence curves explains why the pairing works in practice. YOLOv9 maintains a broad operating plateau that keeps alerts stable as conditions vary, while Faster R-CNN exhibits sharper high-precision behavior that is valuable in cluttered environments. The remaining weakness on the Destroyed Infrastructure class identifies a concrete path for improvement without changing the overall design. Taken together, the results show that the proposed pipeline improves speed, accuracy, and decision readiness for search and rescue, hazardous-area monitoring, and infrastructure assessment.

Table 1 summarizes the comparative behavior of the two models within the UAV–UGV system.

**Table 1.** Comparative summary of YOLOv9 and Faster R-CNN in the proposed UAV–UGV pipeline.

| Metric | YOLOv9 | Faster R-CNN |
|---|---|---|
| mAP@0.5 | 0.969 | 0.979 |
| F1-score | 0.95 | 0.95 |
| Inference speed | 41.7 FPS (∼24× Faster) | 1.72 FPS |
| Model Size | ∼15 MB | ∼165 MB |
| Specific Strength | Fast, real-time scanning | Precise validation |

## 7. Future Work

Future work could examine system-level behavior beyond detector scores, including the fraction of aerial cues confirmed by the ground robot, the fraction rejected as false alarms, and the time from first cue to confirmation, so that end-to-end performance can be compared across sites. It would also be valuable to report mAP@[.5:.95] and AP75, together with confidence-calibration analyses, to emphasize localization quality under stricter IoU thresholds and to align predicted scores with empirical precision.

Another promising direction is to accelerate ground-level verification. Structured pruning, quantization, layer fusion, device-specific compilation, and lighter two-stage backbones could be assessed to raise throughput on the UGV while preserving precision, with accuracy–latency Pareto curves reported on target hardware. Robustness may be improved through multi-sensor fusion, for example, LiDAR or stereo depth for boundary stabilization and thermal or gas sensing for low-light and chemical events, with feature- or decision-level fusion that remains stable when communications degrade.

Generalization could be strengthened by tiling and multi-scale sampling for very large structures, by targeted augmentation for clutter and occlusion, and by continual or federated learning that adapts models to new sites without centralizing sensitive data. Finally, controlled field trials in representative environments, including night, haze, rain, GPS-denied areas, and limited-bandwidth links, would help quantify operational constraints, operator workload, and the pipeline's reliability under realistic deployment conditions.

**Author Contributions:** Conceptualization, A.H. and Z.H.; methodology, A.H. and Z.H.; software, A.H.; validation, A.H., Z.H. and M.H.; formal analysis, A.H.; investigation, A.H.; resources, A.H. and Z.H.; data curation, A.H.; writing—original draft preparation, A.H.; writing—review and editing, A.H., Z.H. and M.H.; visualization, A.H.; supervision, Z.H.; and project administration, Z.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available from the corresponding author upon request.

**Acknowledgments:** GenAI (ChatGPT 5.0) was used for English language polishing. All scientific content, results, and conclusions were produced and verified by the authors.

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

## References

1. Lyu, M.; Zhao, Y.; Huang, C.; Huang, H. Unmanned aerial vehicles for search and rescue: A survey. *Remote Sens.* **2023**, *15*, 3266. [CrossRef]
2. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. Yolov9: Learning what you want to learn using programmable gradient information. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2024.
3. Munasinghe, I.; Perera, A.; Deo, R.C. A comprehensive review of UAV-UGV collaboration: Advancements and challenges. *J. Sens. Actuator Netw.* **2024**, *13*, 81. [CrossRef]
4. Dorigo, M.; Floreano, D.; Gambardella, L.M.; Mondada, F.; Nolfi, S.; Baaboura, T.; Birattari, M.; Bonani, M.; Brambilla, M.; Brutschy, A.; et al. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robot. Autom. Mag.* **2013**, *20*, 60–71. [CrossRef]
5. Neubauer, K.; Bullard, E.; Blunt, R. *Collection of Data with Unmanned Aerial Systems (UAS) for Bridge Inspection and Construction Inspection*; United States Department of Transportation, Federal Highway Administration: Washington, DC, USA, 2021.
6. Tijtgat, N.; Van Ranst, W.; Goedeme, T.; Volckaert, B.; De Turck, F. Embedded real-time object detection for a UAV warning system. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*; IEEE: Piscataway, NJ, USA, 2017.
7. Huda, S.A.; Moh, S. Survey on computation offloading in UAV-enabled mobile edge computing. *J. Netw. Comput. Appl.* **2022**, *201*, 103341. [CrossRef]
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]

9. Rizk, Y.; Awad, M.; Tunstel, E.W. Cooperative heterogeneous multi-robot systems: A survey. *ACM Comput. Surv.* **2019**, *52*, 1–31. [CrossRef]
10. Schweim, A.; Zager, M.; Schweim, M.; Fay, A.; Horn, J. Unmanned vehicles on the rise: A review on projects of cooperating robot teams. *at-Automatisierungstechnik* **2024**, *72*, 3–14. [CrossRef]
11. Ismail, Z.H.; Sariff, N.; Hurtado, E.G. A survey and analysis of cooperative multi-agent robot systems: Challenges and directions. *Appl. Mob. Robot.* **2018**, *5*, 8–14.
12. de Castro, G.G.; Santos, T.M.; Andrade, F.A.; Lima, J.; Haddad, D.B.; Honório, L.D.M.; Pinto, M.F. Heterogeneous multi-robot collaboration for coverage path planning in partially known dynamic environments. *Machines* **2024**, *12*, 200. [CrossRef]
13. Madridano, Á.; Al-Kaff, A.; Flores, P.; Martín, D.; de la Escalera, A. Software architecture for autonomous and coordinated navigation of UAV swarms in forest and urban firefighting. *Appl. Sci.* **2021**, *11*, 1258. [CrossRef]
14. Bayındır, L. A review of swarm robotics tasks. *Neurocomputing* **2016**, *172*, 292–321. [CrossRef]
15. Skorobogatov, G.; Barrado, C.; Salamí, E. Multiple UAV systems: A survey. *Unmanned Syst.* **2020**, *8*, 149–169. [CrossRef]
16. Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. [CrossRef]
17. Yamauchi, B. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97)*; IEEE: Piscataway, NJ, USA, 1997.
18. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: A review. *IEEE J. Ocean. Eng.* **2013**, *39*, 131–149. [CrossRef]
19. Burgard, W.; Moors, M.; Stachniss, C.; Schneider, F.E. Coordinated multi-robot exploration. *IEEE Trans. Robot.* **2005**, *21*, 376–386. [CrossRef]
20. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2017**, *32*, 1309–1332. [CrossRef]
21. Tranzatto, M.; Mascarich, F.; Bernreiter, L.; Godinho, C.; Camurri, M.; Khattak, S.; Dang, T.; Reijgwart, V.; Johannes, L.; Wisth, D.; et al. Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA subterranean challenge. *arXiv* **2022**, arXiv:2201.07067. [CrossRef]
22. Berger, G.S.; Teixeira, M.; Cantieri, A.; Lima, J.; Pereira, A.I.; Valente, A.; Castro, G.G.R.d.; Pinto, M.F. Cooperative heterogeneous robots for autonomous insects trap monitoring system in a precision agriculture scenario. *Agriculture* **2023**, *13*, 239. [CrossRef]
23. Zhang, Z.; Wu, J.; Dai, J.; Ying, J.; He, C. Cooperative tactical planning method for UAV formation. In *2020 39th Chinese Control Conference (CCC)*; IEEE: Piscataway, NJ, USA, 2020.
24. Shaferman, V.; Shima, T. Unmanned aerial vehicles cooperative tracking of moving ground target in urban environments. *J. Guid. Control. Dyn.* **2008**, *31*, 1360–1371. [CrossRef]
25. Wang, Z.; Li, J.; Li, J.; Liu, C. A decentralized decision-making algorithm of UAV swarm with information fusion strategy. *Expert Syst. Appl.* **2024**, *237*, 121444. [CrossRef]
26. Fan, B.; Li, Y.; Zhang, R.; Fu, Q. Review on the technological development and application of UAV systems. *Chin. J. Electron.* **2020**, *29*, 199–207. [CrossRef]
27. Liu, C.; Zhao, J.; Sun, N. A review of collaborative air-ground robots research. *J. Intell. Robot. Syst.* **2022**, *106*, 60. [CrossRef]
28. Cieslewski, T.; Choudhary, S.; Scaramuzza, D. Data-efficient decentralized v isual SLAM. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2018.
29. Alqudsi, Y.; Aydemir, E.; Kerdığe, B. A Comprehensive Review of Aerial Robots for Search and Rescue Operations. *Preprints* **2025**. [CrossRef]
30. Kaliszewski, M.; Włodarski, M.; Młyńczak, J.; Jankiewicz, B.; Auer, L.; Bartosewicz, B.; Liszewska, M.; Budner, B.; Szala, M.; Schneider, B.; et al. The multi-gas sensor for remote UAV and UGV missions—Development and tests. *Sensors* **2021**, *21*, 7608. [CrossRef]
31. Christie, G.; Shoemaker, A.; Kochersberger, K.; Tokekar, P.; McLean, L.; Leonessa, A. Radiation search operations using scene understanding with autonomous UAV and UGV. *J. Field Robot.* **2017**, *34*, 1450–1468. [CrossRef]
32. Chen, H.; Hou, L.; Zhang, G.K.; Wu, S. Using context-guided data augmentation, lightweight CNN, and proximity detection techniques to improve site safety monitoring under occlusion conditions. *Saf. Sci.* **2023**, *158*, 105958. [CrossRef]
33. Shahar, F.S.; Sultan, M.T.H.; Nowakowski, M.; Łukaszewicz, A. UGV-UAV integration advancements for coordinated missions: A review. *J. Intell. Robot. Syst.* **2025**, *111*, 69. [CrossRef]
34. Kucharczyk, M. Pre- and Post-Disaster Remote Sensing with Drones for Supporting Disaster Management. Doctoral Thesis, University of Calgary, Calgary, AB, Canada, 2023.
35. Zhang, Y.; Yan, H.; Zhu, D.; Wang, J.; Zhang, C.-H.; Ding, W.; Luo, X.; Hua, C.; Meng, M.Q.-H. Air-ground collaborative robots for fire and rescue missions: Towards mapping and navigation perspective. *arXiv* **2024**, arXiv:2412.20699. [CrossRef]
36. Daoud, E.; Khalil, N.; Gaedke, M. Implementation of a one-stage object detection solution to detect counterfeit products marked with a quality mark. *IADIS Int. J. Comput. Sci. Inf. Syst.* **2022**, *17*, 33–49.
37. Nurkarim, W.; Wijayanto, A.W. Building footprint extraction and counting on very high resolution satellite imagery using object detection deep learning framework. *Earth Sci. Inform.* **2023**, *16*, 515–532. [CrossRef]

38. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016.
40. Li, J.; Feng, Y.; Shao, Y.; Liu, F. IDP-YOLOV9: Improvement of object detection model in severe weather scenarios from drone perspective. *Appl. Sci.* **2024**, *14*, 5277. [CrossRef]
41. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018.
42. Tan, M.; Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2019.
43. Li, Q.; Xu, X.; Guan, J.; Yang, H. The improvement of Faster-RCNN crack recognition model and parameters based on attention mechanism. *Symmetry* **2024**, *16*, 1027. [CrossRef]
44. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020.
45. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
46. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 12697–12705.
47. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2012; pp. 3354–3361.
48. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2020; pp. 11621–11631.
49. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 2961–2969.
50. Poudel, R.P.; Liwicki, S.; Cipolla, R. Fast-SCNN: Fast semantic segmentation network. *arXiv* **2019**, arXiv:1902.04502. [CrossRef]
51. Campos, C.; Elvira, R.; Gómez Rodríguez, J.J.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
52. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 Year, 1000 km: The Oxford RobotCar Dataset. *Int. J. Robot. Res.* **2017**, *36*, 3–15. [CrossRef]


**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.