mathematics

MDPI

Article

LightSeek-YOLO: A Lightweight Architecture for Real-Time Trapped Victim Detection in Disaster Scenarios

Xiaowen Tian [1,*], Yubi Zheng [1], Liangqing Huang [2], Rengui Bi [1], Yu Chen [1], Shiqi Wang [1] and Wenkang Su [3,*]

[1] College of Physics, Mechanical and Electrical Engineering, Jishou University, Jishou 416000, China; zheng120@stu.jsu.edu.cn (Y.Z.); birengui@jsu.edu.cn (R.B.); 2023405389@stu.jsu.edu.cn (Y.C.); 2023405380@stu.jsu.edu.cn (S.W.)
[2] College of Computer Science and Engineering, Jishou University, Jishou 416000, China; huanglq@stu.jsu.edu.cn
[3] School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China
* Correspondence: tianxw5@jsu.edu.cn (X.T.); swk1004@gzhu.edu.cn (W.S.)

Abstract

Rapid and accurate detection of trapped victims is vital in disaster rescue operations, yet most existing object detection methods cannot simultaneously deliver high accuracy and fast inference under resource-constrained conditions. To address this limitation, we propose the LightSeek-YOLO, a lightweight, real-time victim detection framework for disaster scenarios built upon YOLOv11. Our LightSeek-YOLO integrates three core innovations. First, it employs HGNetV2 as the backbone, whose HGStem and HGBlock modules leverage depthwise separable convolutions to markedly reduce computational cost while preserving feature extraction. Secondly, it introduces Seek-DS (Seek-DownSampling), a dual-branch downsampling module that preserves key feature extrema through a MaxPool branch while capturing spatial patterns via a progressive convolution branch, thereby effectively mitigating background interference. Third, it incorporates Seek-DH (Seek Detection Head), a lightweight detection head that processes features through a unified pipeline, enhancing scale adaptability while reducing parameter redundancy. Evaluated on the common C2A disaster dataset, LightSeek-YOLO achieves 0.478 AP@small for small-object detection, demonstrating strong robustness in challenging conditions such as rubble and smoke. Moreover, on the COCO, it reaches 0.473 mAP@[0.5:0.95], matching YOLOv8n while achieving superior computational efficiency through 38.2% parameter reduction and 39.5% FLOP reduction, and achieving 571.72 FPS on desktop hardware, with computational efficiency improvements suggesting potential for edge deployment pending validation.

Keywords: lightweight object detection; disaster scenarios; trapped victim detection; YOLOv11

MSC: 68T05

check for updates

Academic Editor: Manuel G. Forero

Received: 27 August 2025
Revised: 28 September 2025
Accepted: 6 October 2025
Published: 9 October 2025

Citation: Tian, X.; Zheng, Y.; Huang, L.; Bi, R.; Chen, Y.; Wang, S.; Su, W. LightSeek-YOLO: A Lightweight Architecture for Real-Time Trapped Victim Detection in Disaster Scenarios. *Mathematics* **2025**, *13*, 3231. https://doi.org/10.3390/math13193231

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Globally, disasters are occurring with increasing frequency and severity, inflicting extensive damage on human society and the ecological environment [1]. The inherent randomness, dynamics, and destructiveness of disasters often create highly complex affected areas, posing significant challenges to rescue operations and amplifying danger. Disaster events generally unfold in four stages: pre-disaster warning, disaster assessment, disaster response, and post-disaster recovery [2]. Secondary hazards such as building

collapses following typhoons or debris flows after heavy rains can exacerbate casualties and losses, while further complicating rescue efforts. Thus, timely and accurate access to critical information is essential for effective emergency response, personnel rescue and post-disaster reconstruction.

Building upon this critical need for timely disaster response, object detection technology emerges as a pivotal solution. Object detection, a core component in computer vision, has substantial potential in disaster scenarios. In the aftermath of earthquakes, detection algorithms can help rescuers rapidly locate trapped victims, thereby enhancing operational efficiency. In forest fires, they can identify ignition points and track fire spread to support firefighting strategies. In floods, they can detect submerged regions and stranded targets to optimize resource allocation. Intelligent object detection can therefore markedly improve both the speed and accuracy of disaster response, reducing casualties and property damage. Given the need for rapid and precise identification in such complex environments, developing specialized object detection methods for disaster scenarios has become a pressing research priority.

Despite these promising applications, several critical challenges persist in disaster-oriented object detection. One key issue is balancing real-time performance and detection accuracy, particularly for UAV platforms with limited computational resources. For example, Dong et al. developed a lightweight YOLOv3-MobileNet human detection model using pruning based on sensitivity analysis [3,4]. This approach achieved reduced model parameters but compromised detection accuracy. Romero et al. addressed this trade-off using a knowledge distillation framework [5], preserving accuracy while lowering complexity, though at the expense of a more complex training process. Small-object detection is another major challenge. Ma et al. proposed a method for detecting small trapped victims in aerial imagery through a static-dynamic bounding box weighted fusion [6], which combines YOLOv4 features [7] with LiteFlowNet3 motion cues [8] to reduce missed detections in high-altitude scenarios, albeit at the expense of increased computational demand. Focusing further on the detection of small-scale and severely occluded objects in complex disaster environments, Hao et al. proposed a YOLOv5-based [9] approach with a hybrid-domain attention mechanism and feature reuse [10], improving detection but struggling under low-light conditions. Liu et al. enhanced small-object accuracy using a feature pyramid network with multi-layer feature fusion and adaptive feature selection [11], but it consumes substantial memory resources. The integration of attention mechanisms, particularly Transformer-based models, has shown promise. Y. Chen et al. [12] introduced the TFSANet architecture, within which a Transformer-based fusion model and a dynamic mechanism relying on selective kernel convolution are embedded. This design endows the TFSANet with the capacity to effectively address challenges associated with low-visibility conditions (including smoke and torrential rain) and target scale variations in post-disaster environments. Notwithstanding, a notable limitation of this architecture is its tendency to miss the detection of densely distributed targets. Concurrently, Z. Chen et al. [13] incorporated the self-adaptive characteristic aggregation fusion (SACAF) attention module, implemented a fusion strategy, and carried out optimizations on the YOLO network. Through these efforts, the resulting model is able to comprehensively preserve features and exhibits robust generalization capabilities. However, it is hampered by a relatively slow detection speed, and there is a high likelihood of it failing to detect extremely small targets ($<8 \times 8$ pixels).

Multi-modal fusion and ensemble methods have also advanced the field. The UGEN framework [14] combines GAN-assisted semantic segmentation with multi-detector ensembles, improving accuracy but with high system complexity. In this regard, Hou et al. [15] introduced the self-supervised difference contrast learning framework (Self-DCF), which is specifically designed for label-free change detection in remote-sensing imagery. By leverag-

ing self-supervised learning techniques, this framework effectively mitigates the challenge of substantial annotation costs associated with datasets. Moreover, it exhibits remarkable robustness. Nevertheless, it is encumbered by two notable drawbacks: high computational expenses and the dependence of image translation processes on noise. J. Zhu et al. [16] explored cross-modal knowledge transfer via domain adaptation, boosting generalization across sensor modalities. On the other hand, several works have focused on domain-specific optimizations. For example, Q. Zhu et al.'s YOLOv7-CSAW [17] improved small-object detection in marine environments using C2f modules [18] and parameter-free attention [19], but with limited generalization. Li et al. [20] put forward a Two-tier Submodel Partition Framework (TSPF) grounded in a two-layer federated learning architecture, which is designed for forest fire detection. Nevertheless, this framework is deficient in an adaptive parameter-tuning mechanism. Chen et al. [21] created a weather-adaptive framework for rain, snow and fog, improving robustness via adaptive modules and data augmentation. In addition, data augmentation has also been explored. Zhu et al. [22] used GANs to generate synthetic disaster images, expanding training datasets, though realism remains imperfect. Gatys et al. [23] applied style transfer for cross-scenario domain adaptation, improving adaptability in new environments.

Despite these advances, several critical gaps persist. First, in low-resource rescue environments, achieving efficient inference without compromising accuracy remains difficult, particularly for UAV deployment. Second, complex backgrounds—debris, smoke and lighting variations—complicate feature extraction, with conventional downsampling prone to losing essential details, especially for small targets like trapped victims. Third, multiscale target distributions exacerbate parameter redundancy in existing detection heads, limiting adaptability under lightweight constraints.

These limitations collectively underscore three fundamental challenges requiring systematic resolution: computational efficiency under resource constraints, robustness against complex background interference, and adaptability across target distributions. To systematically address these three core limitations, we present the LightSeek-YOLO, a lightweight architecture based on YOLOv11 [24] for real-time victim detection in disaster scenarios, incorporating three key innovations:

1. Efficiency under limited resources—using HGNetV2 [25] as the backbone, with depth-wise separable convolution-based HGStem and HGBlock modules, reducing computational cost while preserving feature extraction.
2. Robustness to complex backgrounds—introducing the Seek-DS dual-branch downsampling module, combining MaxPool-based extrema preservation with progressive convolution-based spatial pattern extraction, reducing information loss in cluttered or smoky environments.
3. Scalable, low-redundancy detection—designing the Seek-DH lightweight multiscale detection head that processes features through a unified pipeline, enhancing scale adaptability while reducing parameter redundancy.

The experimental results demonstrate that LightSeek-YOLO achieves an excellent balance of accuracy and efficiency. In the COCO dataset, it delivers a competitive mAP@[0.5:0.95] of 0.473, matching the performance of YOLOv8n (0.473) and approaching that of its baseline, YOLOv11n (0.481). Crucially, this performance is attained with a significant reduction in computational overhead, requiring only 1.86M parameters and 4.9 GFLOPs—a reduction of 27.8% and 22.2% compared to YOLOv11n, respectively. The model also achieves a high inference speed of 571.72 FPS on a standard GPU, demonstrating its capability for real-time processing. Moreover, on the specialised C2A disaster dataset, the model achieves an AP@small of 0.478, thereby confirming its robust performance and particular efficacy in detecting small targets amid challenging conditions such as

debris and smoke. These results collectively validate efficiency improvements that suggest deployment potential on edge platforms, requiring comprehensive edge device validation in disaster scenarios.

Our method targets these limitations through systematic architectural improvements. The remaining sections of this paper are structured as follows. In Section 2, we provide a concise review of related work. The overall architecture of the proposed method is introduced in Section 3. In Section 4, we present the experimental results and conduct a discussion. Finally, Section 7 summarizes the conclusions and outlines prospects.

2. Related Work

In the present paper, we put forward the LightSeek-YOLO model for real-time trapped victim detection tasks in disaster scenarios. This task is intimately associated with the design of the lightweight backbone network, efficient downsampling strategies, and multiscale detection head optimization. To tackle the three core challenges in disaster scenarios (limited computational resources, complex background interference, and multiscale object detection), existing research has proposed various solutions. However, these approaches are deficient in systematic lightweight design. The design of the lightweight backbone network is intended to reduce computational complexity so as to ensure efficient inference in resource-constrained rescue environments. Efficient downsampling strategies must preserve key information while reducing feature map resolution and addressing complex background interference in disaster scenarios. Multiscale detection head optimization requires effectively handling trapped victim object detection at different scales under parameter-constrained conditions. In this section, we briefly review the existing research on these topics.

2.1. Lightweight Backbone Networks

The research regarding lightweight backbone networks primarily concentrates on reducing computational complexity and parameter count while maintaining feature extraction capabilities. The MobileNet series [26] substantially decreases computational cost through depth-wise separable convolution. Conversely, it has limited feature representation capability in complex scenarios. EfficientNet [27] achieves a good balance between accuracy and efficiency through a compound scaling strategy, but is difficult to optimize for specific application scenarios. GhostNet [28] generates additional feature maps through Ghost modules, substantially reducing parameter count while maintaining performance. Nonetheless, it exhibits limitations in small-object detection. The latest HGNetV2 improves feature extraction efficiency through enhanced HGBlock and HGStem modules. Its hardware-aware lightweight design provides new possibilities for real-time detection in disaster scenarios.

2.2. Downsampling Strategies

Building upon lightweight backbone design, the choice of downsampling strategy is equally critical. Downsampling, constituting a pivotal operation within deep learning networks, directly impacts feature preservation and computational efficiency. Traditional max pooling (MaxPool) can retain local maximum values but loses spatial detail information. On the other hand, strided convolution achieves downsampling through learnable parameters yet tends to be susceptible to noise interference in complex backgrounds. SPP (Spatial Pyramid Pooling) [29] captures features of different receptive fields through pooling. Notwithstanding, it has a large computational overhead. In recent research, SPD-Conv [30] achieves lossless downsampling through space-to-depth conversion. However,

current methods still lack a targeted design for occlusion and deformation problems that are specific to disaster scenarios.

2.3. Detection Head Designs

Complementing efficient downsampling, detection head design directly impacts the final performance. The detection head, being a pivotal component of object detection networks, directly dictates detection accuracy and inference efficiency. Traditional single-stage detectors, such as the YOLO series [31], utilize convolutional layers to construct detection head. These heads are structurally simple but suffer from parameter redundancy. In contrast, Feature Pyramid Networks (FPNs) [32] fuse features of different scales via top-down paths and lateral connections, yet this increases computational complexity. BiFPN [33] enhances efficiency through bidirectional feature fusion and weighted fusion mechanisms. Nonetheless, it still has parameter redundancy issues in lightweight scenarios. The Decoupled Head [34] significantly improves detection performance by separating classification and regression tasks, at the expense of increased network complexity. YOLOv11 further boosts detection accuracy and inference efficiency through enhanced C3k2 modules and an optimized detection head structure. However, there is still scope for targeted optimization in trapped victim detection for disaster scenarios.

2.4. Critical Gaps in Disaster Detection Systems

Notwithstanding the significant progress that has been made in lightweight network design, downsampling optimization, and detection head improvement, several limitations remain. Specifically, existing lightweight methods have inadequate robustness in complex disaster scenario backgrounds, making it difficult to balance accuracy and efficiency. Moreover, traditional downsampling strategies are prone to losing key information when handling occlusion and deformation issues in disaster scenarios. Furthermore, existing detection head designs lack specialized optimization for trapped victim detection and suffer from parameter redundancy in target processing. To address these problems, we propose the LightSeek-YOLO method. This approach aims to achieve an optimal balance between accuracy and efficiency in trapped victim detection for disaster scenarios through the collaborative design of the HGNetV2 backbone network, the Seek-DS dual-branch downsampling module, and the Seek-DH lightweight detection head. Next, we will provide an in-depth introduction to the design principles and implementation details of each module.

3. Proposed Method

To address the limitations of existing methods discussed in Section 2, we propose LightSeek-YOLO, a lightweight architecture built upon the YOLOv11 framework. The overall architecture is shown in Figure 1. The method tackles core challenges in disaster scenarios through three key modules: (1) the HGNetV2 backbone network, which is designed to overcome limited computational resources; (2) the Seek-DS downsampling module, which mitigates complex background interference; and (3) the Seek-DH detection head, which is optimized for object detection. Each module is tailored to the characteristics of disaster scenarios, achieving an effective balance between accuracy and efficiency. The following subsections detail the design principles and implementation of each module.

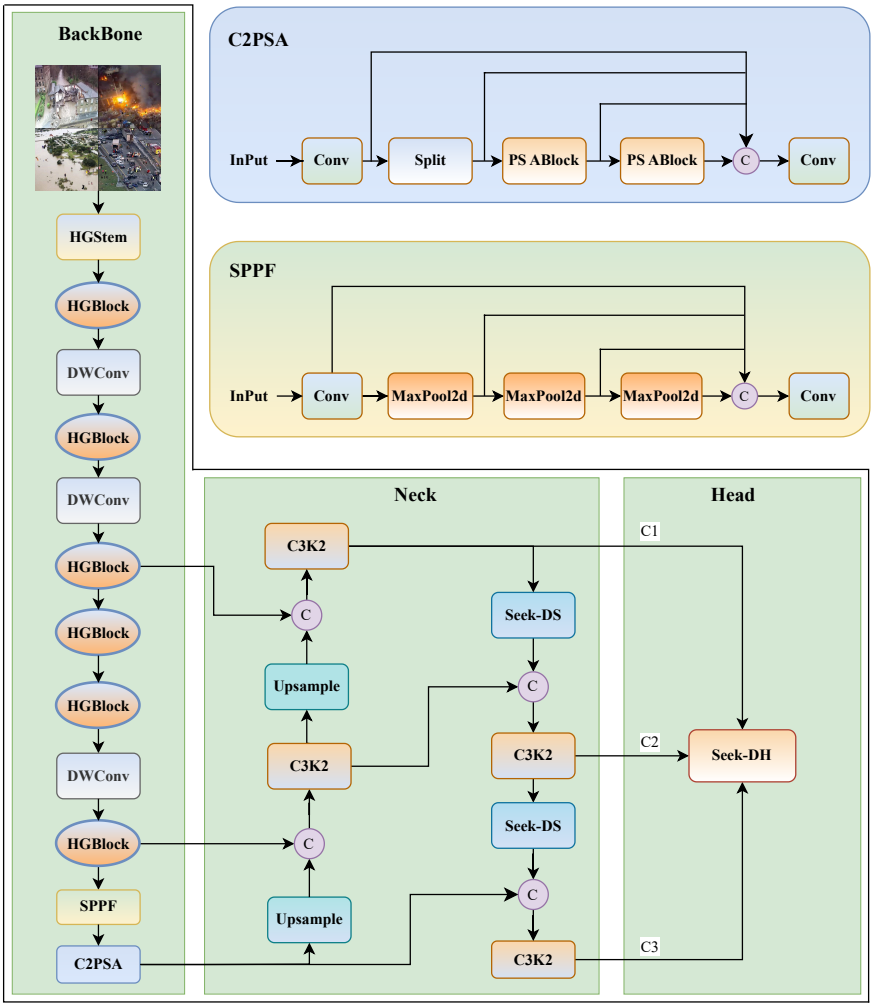


Figure 1. Overall Network Architecture of LightSeek-YOLO. As depicted in the figure, the overall network structure of LightSeek-YOLO is presented, comprising five core components: the input layer, the HGNetV2 backbone network, the Seek-DS dual-branch downsampling module, the feature fusion network, and the Seek-DH lightweight detection head. The HGNetV2 backbone network is responsible for feature extraction. Subsequently, the Seek-DS module retains key information during downsampling through its dual-branch collaborative design. The feature fusion network integrates features of different scales, and finally, the Seek-DH detection head outputs the location and category information of trapped victims. The entire architecture adopts end-to-end training and has been specifically optimized for real-time detection requirements in disaster scenarios.

3.1. HGNetV2

To address computational efficiency requirements while maintaining feature extraction capability, we select HGNetV2 as our backbone network. During the task of detecting trapped victims in disaster scenarios, the target typically exhibits characteristics such as variations, severe occlusions and complex backgrounds, which impose higher demands on the feature extraction capabilities of backbone networks. While traditional ResNet [35] backbone networks mitigate the vanishing gradient problem through residual connections, their hierarchical feature fusion efficiency is relatively low. Additionally, as network depth increases, computational complexity rises sharply. Especially when processing high-resolution disaster scene images, ResNet struggles to effectively capture features with a single convolution operation, leading to degraded performance in small object detection.

Inspired by RT-DETR [25], we adopt the lightweight network module HGNetV2 as the backbone for feature extraction (Figure 2). Based on the single-stage object detection framework of the YOLO series, as shown in Figure 2, our model efficiently extracts features

and performs downsampling by using the HGStem block in the initial stage. Specifically, compared to the input layer of the original network, the HGStem block improves the computational efficiency of the network by reducing the dimension of the initial feature map and eliminating redundant parameters. The specific workflow is illustrated in Figure 3a. As for the HGBlock module, it plays a crucial role in the HGNetV2 backbone, with its overall structure shown in Figure 3b. Specifically, in the HGNetV2 architecture, we stack HGBlock modules to ensure that the features extracted from the network can fuse information from different scales and depths. Therefore, the design of HGBlock optimizes the network's processing capabilities, enabling it to more effectively handle scale changes in images. This is particularly useful for identifying trapped victims of different sizes in disaster scenarios.

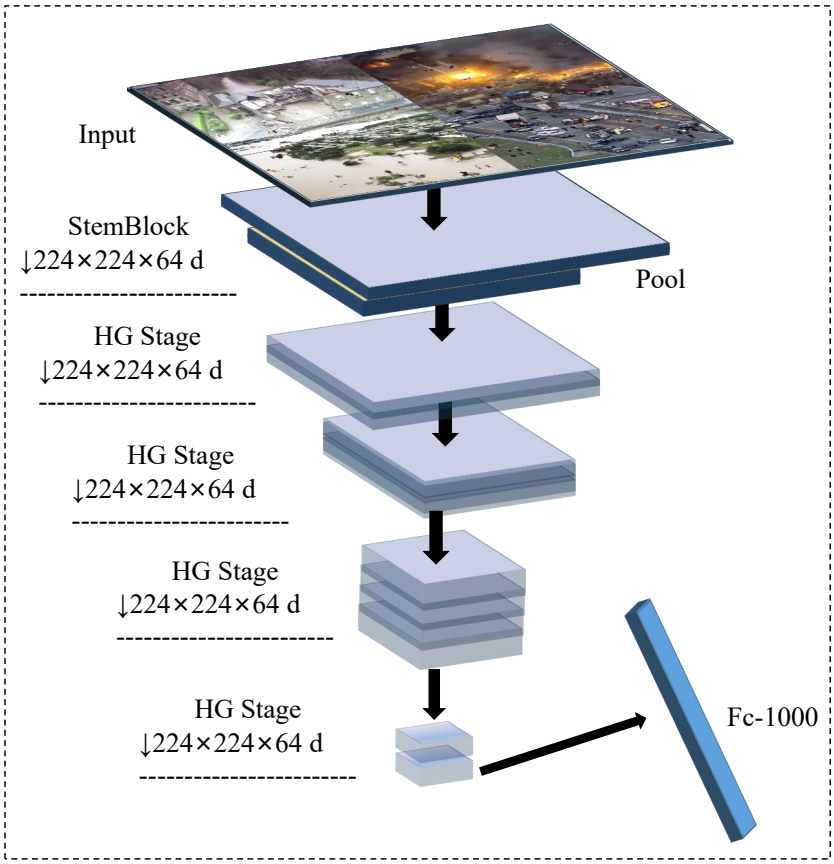


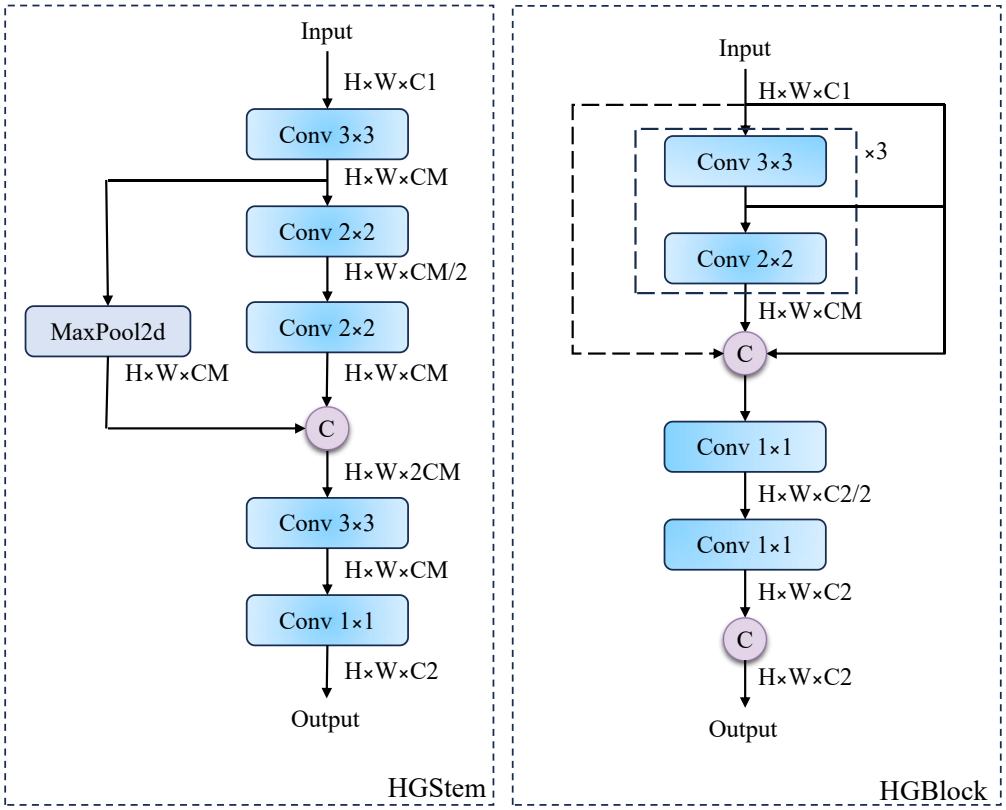
Figure 2. HGNetV2 network architecture diagram.

Through the deployment of depthwise separable convolution architecture in HGStem and HGBlock modules, the HGNetV2 backbone network substantially reduces computational complexity while preserving feature extraction capabilities. Compared to traditional ResNet architecture, this design demonstrates a superior balance of efficiency and accuracy in disaster scenarios for trapped victim detection tasks, providing a reliable technical foundation for real-time rescue applications.

3.2. Seek-DS

For optimal information preservation during feature fusion, we introduce Seek-DS in the neck architecture. In the neck design of traditional object detection networks, standard convolution modules (Conv) typically achieve downsampling through a single convolution operation (e.g., a 3×3 convolution kernel with a stride of 2). However, in disaster scenarios, complex background interference (such as debris and smoke obstruction) and significant variations in target scale can cause single-path downsampling to lose critical detail information, thereby affecting the accurate localization of small targets (such as

trapped victims). To address the feature loss issues caused by complex background interference in disaster scenarios using traditional downsampling methods, we propose the Seek-DS (Seek-DownSampling) dual-branch heterogeneous collaborative downsampling module. This module integrates a maximum pooling branch that preserves extrema with a progressive convolution branch, maximizing the retention of critical information while reducing the resolution of feature maps.



(**a**) HGStem (**b**) HGBlock

Figure 3. (**a**) The network workflow of HGStem; (**b**) The network workflow of HGBlock.

As shown in Figure 4, the Seek-DS module adopts a dual-branch parallel processing architecture to achieve efficient feature downsampling through two complementary paths. The first branch is the MaxPool extreme value retention path, which retains the maximum activation value in a local region through maximum pooling operations, while also using 1×1 convolution for feature preprocessing to boost learnability. The second path is the progressive convolution path, which uses a cascaded 1×1–3×3 convolution structure to achieve progressive feature extraction from the channel dimension to the spatial dimension. Ultimately, these two complementary paths work collaboratively to achieve efficient feature downsampling.

The traditional MaxPool operation divides the feature map into multiple non-overlapping local regions based on the predefined pooling window size $k \times k$ and stride s, with each window containing k^2 elements. During the extreme value calculation phase, the maximum value operation is strictly applied to each element within the window, mathematically expressed as follows:

$$Y_{i,j,c} = \max X_{i \cdot s+p, j \cdot s+q, c}. \quad (1)$$

Here, i and j are the coordinates of the feature map. c is the channel index. s is the stride, which determines the starting position of the pooling window. p and q are the

relative coordinates within the pooling window (with the pooling window size being $k \times k$), and their range is from 0 to $k - 1$.

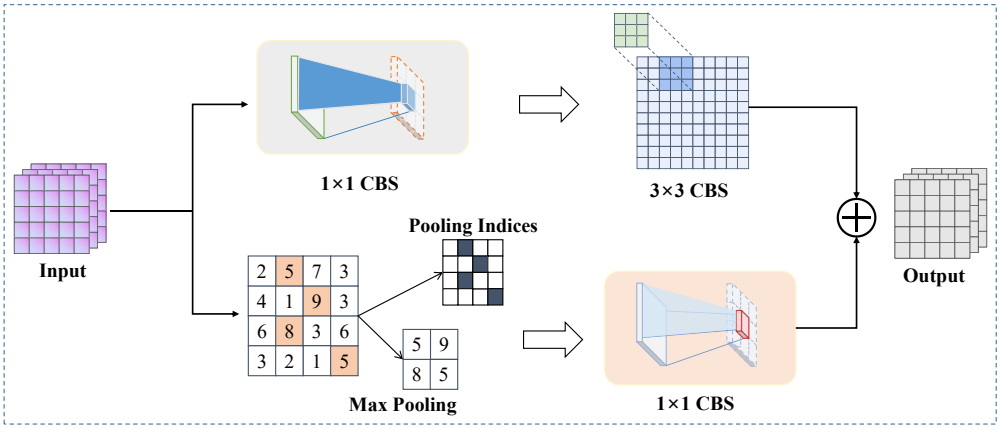


Figure 4. Architecture diagram of the Seek-DS dual-branch heterogeneous collaborative downsampling module.

After pooling processing, the output feature map size is reduced according to the following formula:

$$H_{out} = \left\lfloor \frac{H + 2 * p - k}{s} \right\rfloor + 1, W_{out} = \left\lfloor \frac{W + 2 * q - k}{s} \right\rfloor + 1. \quad (2)$$

Among them, H and W are the height and width of the feature map, respectively; k represents the size of the convolution kernel; s represents the stride; and p indicates the padding number (when the "same" padding mode is adopted).

To address the issue of detail loss in the MaxPool operation during feature downsampling, the first branch of the Seek-DS module innovatively introduces a cascaded 1×1 convolution structure to establish a multi-level feature compensation mechanism. For specific implementation, in the original MaxPool path, we first perform 1×1 convolution pre-processing before the pooling operation:

$$Y_{\text{pre}} = W * X + b. \quad (3)$$

Through a learnable weight matrix $W \in \mathbb{R}^{C \times C}$, channel-wise feature reconstruction is performed on the input features. The preprocessed features are then fed into a 2×2 MaxPool operation, where the maximum pooling calculation process is as follows:

$$Y_{pool} = MaxPool\left(Y_{pre}\right) = MaxPool(W * X + b). \quad (4)$$

As for the construction of the second branch in the Seek-DS module, it adopts a progressive spatial aggregation strategy, which achieves multi-level nonlinear feature aggregation through a cascaded 1×1–3×3 convolution structure. In the progressive convolution branch, the input features are first preprocessed via a 1×1 convolution, which performs a linear transformation on the feature map along the channel dimension. The weight matrix $W_1 \in \mathbb{R}^{C/2 \times C \times 1 \times 1}$ (W_1 represents a 1×1 convolutional kernel with a compression ratio of 0.5) is used to reduce the number of channels to half of the original value. Subsequently, Batch Normalization (BN) [36] and the SiLU (Sigmoid Linear Unit) [37] activation function are introduced to introduce nonlinearity:

$$Y_1 = SiLU(BN(W_1 * X)). \quad (5)$$

To compensate for insufficient spatial information interaction in 1×1 convolution, 3×3 convolution is cascaded after 1×1 convolution to construct a composite feature extraction path. The 3×3 convolution plays three key roles here: first, establishing spatial dimension feature correlations through 3×3 local receptive fields, where convolution kernel $W_2 \in \mathbb{R}^{C/2 \times C/2 \times 3 \times 3}$ can effectively capture spatial patterns in local regions; second, implementing regular feature map size reduction through the stride-2 downsampling strategy; and finally, adaptively adjusting feature response intensity at different spatial positions through parameterized weight learning:

$$Y_2 = SiLU(BN(W_2 * Y_1)). \tag{6}$$

The second branch performs three important functions throughout the module: first, it achieves progressive feature extraction from the channel to the spatial dimension through a cascaded 1×1–3×3 structure. Second, it establishes functional complementarity with the MaxPool operation of the first branch, whereby the first branch focuses on retaining feature extrema, while the second branch excels at learning spatial patterns. Third, it provides a diverse feature representation foundation for subsequent feature fusion. The outputs of the two branches are fused through a concatenation operation, ultimately forming a feature representation that combines both detail retention capability and semantic abstraction capability.

In summary, the Seek-DS dual-branch heterogeneous collaborative downsampling module preserves key feature extrema through the MaxPool branch and captures spatial patterns via the progressive convolution branch, thereby effectively addressing complex background interference encountered in disaster scenarios. This design aims to enhance computational efficiency and reduce information loss, making it particularly suitable for handling the occlusion and deformation object detection tasks commonly encountered in disaster scenarios.

3.3. Seek-DH

To eliminate parameter redundancy while maintaining detection capability, we design Seek-DH for the detection head. Traditional YOLO detection heads adopt independent branch architectures, where each scale layer requires a dedicated convolutional network for object detection. Although this design enables scale-specific feature processing, it introduces substantial parameter redundancy. To overcome this limitation, we propose the Seek-DH, a shared detection head that processes multiple-scale features through a unified pipeline, thereby reducing parameter redundancy inherent in conventional designs.

As illustrated in Figure 5, Seek-DH employs a feature processing strategy to establish a unified detection framework capable of handling features across scales simultaneously. As we know, in disaster scenarios, personnel targets exhibit a wide range of scales—distant individuals appear as small targets, while nearby individuals appear as large ones. In contrast, the original YOLO detection head requires separate branches for the P3, P4 and P5 feature maps, a design that not only increases network complexity but also risks fragmenting inter-scale feature information.

To enable unified processing, Seek-DH first preprocesses three features—C1, C2 and C3, i.e., the P3, P4 and P5 feature maps of the backbone. Each one is passed through an independent 1×1 Conv_GN module to project features of varying dimensions into a unified hidden space. This ensures that subsequent operations are performed within a consistent feature space, avoiding information loss from dimensional mismatches. The 1×1 convolution achieves this transformation efficiently while incurring minimal computational cost. After dimensional unification, Seek-DH applies a unified feature processing module composed of two consecutive 3×3 Conv_GN modules (Figure 5). The first em-

ploys depthwise convolution to capture spatial information while controlling parameter growth, and the second further integrates and refines feature representations. Crucially, all scales are processed with identical convolutional parameters, allowing the model to learn cross-scale, universal representations while eliminating the parameter redundancy inherent in scale-specific extractors. Finally, Seek-DH employs the decoupled output strategy for object detection. The processed features are fed into independent Classification (Cls) and Regression (Reg) branches. The classification branch predicts category information through dedicated convolutional layers, outputting class confidence scores, while the regression branch leverages the DFL mechanism to predict bounding boxes with improved localization accuracy. This decoupled design preserves task independence between classification and regression while enabling specialized optimization for each task.

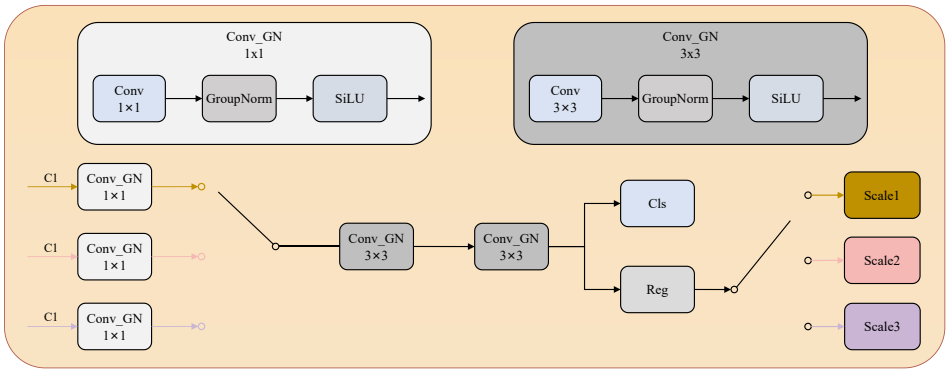


Figure 5. Diagram of Seek-DH: a shared detection head.

Recognizing that features at different scales contribute unequally to detection, Seek-DH incorporates a scale-adaptive adjustment mechanism. As shown in Figure 5, the input features C1, C2 and C3 are first unified in dimension using independent 1×1 Conv_GN modules, then processed jointly through the dual 3×3 Conv_GN feature module. The outputs are subsequently passed through decoupled classification and regression branches to generate preliminary predictions, which are finally refined by the corresponding scale adjustment modules (scaling factors initialized to 1.0), i.e., Scale1, Scale2 and Scale3, to adaptively weight bounding box predictions for each scale. In summary, this streamlined pipeline minimizes parameter redundancy while maintaining robust feature processing, thereby enhancing adaptability for complex personnel detection tasks within disaster environments. In contrast to traditional independent branch designs, our approach combines unified feature processing with a decoupled output strategy and adaptive scaling, ensuring both efficient parameter utilization and balanced detection performance across scales. This makes Seek-DH particularly well-suited for the demands of personnel detection in disaster scenarios.

4. Experiments

We developed a comprehensive experimental validation framework to assess the effectiveness of LightSeek-YOLO in detecting humans during disaster scenarios. The framework encompasses comparative evaluations against existing methods, verification of individual component contributions, and ablation studies.

4.1. Dataset

For evaluation, we employed the C2A dataset [38], a synthetic dataset specifically designed for unmanned aerial vehicle (UAV) search-and-rescue (SAR) operations in disaster environments. This dataset integrates realistic disaster-scene backgrounds with diverse human postures, supplemented by a COCO subset to improve model generalization.

The C2A dataset comprises two primary components:

1. Disaster backgrounds—1345 images representing four types of disasters from the AIDER dataset (fire/smoke, flood, building collapse/ruins and traffic accidents) [39], providing realistic environmental contexts.
2. Human postures—29,732 human instances extracted from 26,675 images in the LSP/MPII-MPHB dataset [40,41], annotated with five postures (bending, kneeling, lying, sitting, upright) to simulate trapped or injured individuals.

The dataset construction pipeline proceeded as follows: human figures were segmented from source images using the U2-Net model [42] with background removal. Instances with a non-zero pixel ratio $\geq$ 2% were retained, and the body regions were cropped. The cropped figures were randomly scaled and composited onto AIDER backgrounds, generating 10,215 synthetic images with bounding box and posture annotations. Following this method, the final dataset contains over 360,000 instances, averaging 20–40 targets per image, with some containing up to 100 targets—effectively simulating high-density scenarios. In terms of scale distribution, 47% of instances are tiny targets (<10 pixels), while 52% range from 10–50 pixels, posing significant challenges for small-object detection. Image resolutions range from 123 $\times$ 152 to 5184 $\times$ 3456 pixels. Combined with five posture types across four disaster contexts, the dataset supports comprehensive fine-grained analysis.

Although synthetic data can present limitations—such as scale or illumination inconsistencies and the absence of dynamic video sequences—the C2A dataset provides a valuable resource for training robust human detection models. By simulating complex disaster conditions, it is particularly suited for SAR applications involving occlusion and small-object detection [38].

4.2. Experimental Configuration

Experiments were conducted on a server equipped with a 15-vCPU Intel(R) Xeon(R) Platinum 8474C processor (Intel Corporation, Santa Clara, CA, USA), an NVIDIA GeForce RTX 4090D GPU and 32 GB of RAM (NVIDIA Corporation, Santa Clara, CA, USA). The implementation was based on Python 3.10 with PyTorch 2.1.0 and CUDA 11.8. Training was performed with a batch size of 32 and 4 worker threads for data loading. The dataset split followed an 8:1:1 train/validation/test ratio to ensure proper model evaluation. To ensure reproducibility, the random seed was set to 0 with deterministic = True. The initial learning rate was set to lr0 = 0.01 with learning rate scheduling to a final value of lrf = 0.01 (1% of the initial rate), using the SGD optimizer with = 0.937 and weight_decay = 0.0005. Loss function weights were configured as box = 7.5, cls = 0.5, dfl = 1.5 to balance different loss components during training. Given the large scale of the C2A dataset, the models were trained for 150 epochs. All hyperparameters were systematically tuned to ensure optimal performance.

4.3. Evaluation Metrics

AP@τ denotes the Average Precision at an Intersection over Union (IoU) threshold τ, which evaluates model performance under relaxed localization criteria (a prediction is considered correct if the overlap between the predicted and ground-truth boxes is at least τ). It is formally defined as

$$AP@\tau = \int_0^1 P(r)\mathrm{d}r. \tag{7}$$

AP@small denotes the average precision for small object detection with an area smaller than 32^2 pixels, following the COCO standard. It evaluates a model's ability to detect small targets and is computed by

$$AP@\text{small} = \int_0^1 P_{small}(r)\mathrm{d}r. \tag{8}$$

mAP@[0.5:0.95] is the primary metric for evaluating overall model accuracy in object detection. It is obtained by computing the Average Precision at 10 Intersection over Union (IoU) thresholds, ranging from 0.5 to 0.95 with a step of 0.05. The final score is the mean of AP values across all thresholds, defined as

$$mAP = \frac{1}{10}\sum_{\tau=0.5}^{0.95} AP@\tau. \tag{9}$$

Other metrics, including FLOPs, parameter count, model size and FPS, are employed to assess the portability of our model, reflecting its computational requirements, memory footprint, and inference efficiency.

4.4. Comparative Experiments (COCO Metrics)

To rigorously evaluate the effectiveness of our YOLOv11-based lightweight human detection method for disaster scenarios, we conducted extensive experiments on the COCO dataset. The comparison models include widely adopted object detectors: YOLO-series models (YOLOv8n and YOLOv11n) representing the latest advances in the YOLO framework, as well as non-YOLO approaches such as Faster R-CNN, SSD, RetinaNet [43] and EfficientDet [44], which cover diverse detection paradigms.

We employed standard COCO evaluation metrics, including mAP and AP at different IoU thresholds, to benchmark our method against these models. As summarized in Table 1, the results clearly demonstrate the relative performance of our approach. By selecting Faster R-CNN, SSD, RetinaNet and EfficientDet as representative non-YOLO baselines, alongside YOLOv8n and YOLOv11n as the most recent lightweight YOLO variants, we provide a comprehensive assessment of our method across accuracy, efficiency and model complexity.

Table 1. Performance comparison of different detection methods on the COCO dataset.

Methods	mAP@ [0.5:0.95]	AP@0.5	AP@0.75	AP@small	Params (M)	FLOPs (G)	Model Size (M)
Faster R-CNN	0.405	0.619	0.443	0.406	60.340	204	472.44
SSD	0.376	0.698	0.361	0.377	24.386	137	193.73
RetinaNet	0.466	0.769	0.472	0.466	18.339	98.565	426.53
EfficientDet	0.491	0.795	0.498	0.492	55.322	198	215.86
Yolov8n	0.473	0.742	0.502	0.478	3.01	8.1	6.11
Yolov11n	0.481	0.748	0.512	0.485	2.58	6.3	5.23
Ours	0.473	0.75	0.5	0.478	1.86	4.9	3.88

Note: "Ours" refers to the complete LightSeek-YOLO model, including Seek-DH detection head, HGNetV2 backbone network and Seek-DS downsampling module.

4.4.1. Comparison with Existing Methods (Non-YOLO Series)

As demonstrated in Table 1, our proposed method achieves competitive detection accuracy while maintaining minimal computational overhead. The model attains an mAP@[0.5:0.95] of 0.473, comparable to high-performance models such as EfficientDet (0.491) and RetinaNet (0.466), while requiring substantially fewer computational resources. Specifically, it demonstrates 16.8% and 25.8% improvements over Faster R-CNN (0.405)

and SSD (0.376), respectively, with significantly reduced parameters and computational cost. For AP@0.5, our model achieves 0.75, approaching EfficientDet's performance (0.795), while substantially outperforming Faster R-CNN (0.619) and SSD (0.698). At the stringent AP@0.75 threshold, our model reaches 0.5, surpassing Faster R-CNN (0.443), SSD (0.361) and RetinaNet (0.472). The small-object detection performance (AP@small = 0.478) validates the method's effectiveness for disaster scenarios involving small-sized targets. The comparison between LightSeek-YOLO and non-YOLO models in disaster scenarios is shown in Figure 6.

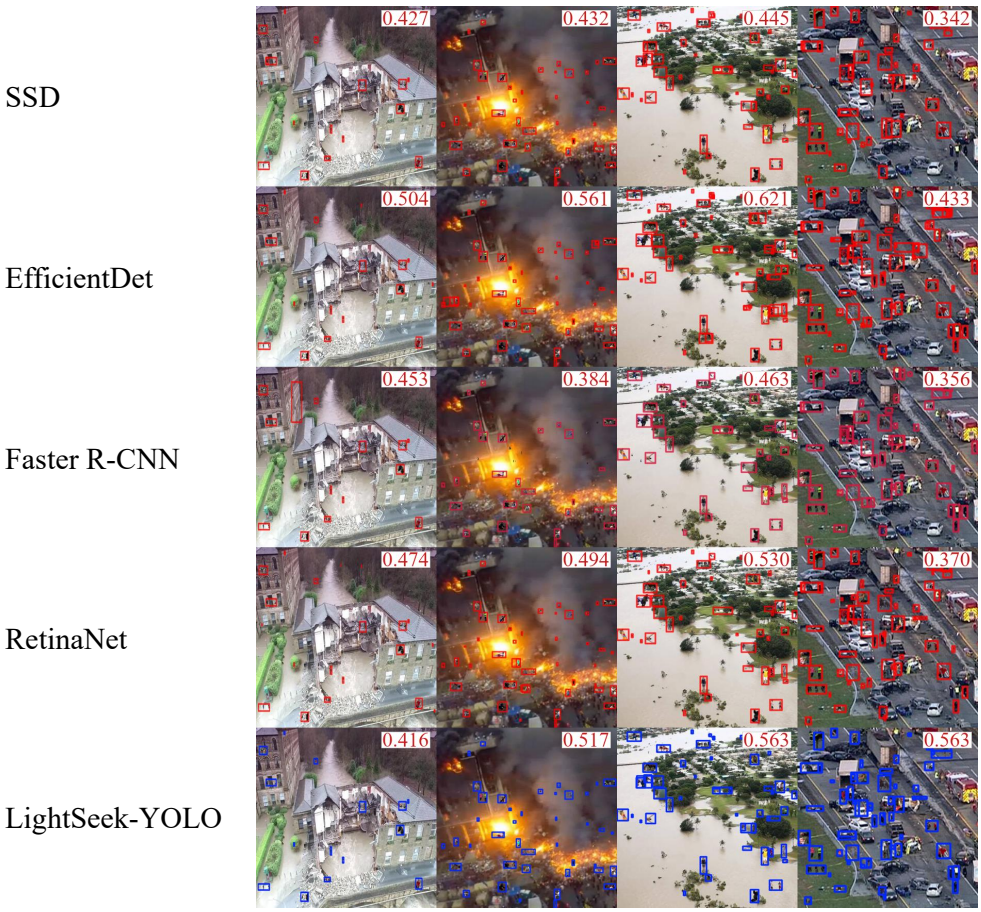


Figure 6. Comparative visualization of LightSeek-YOLO and non-YOLO models on disaster scenarios.

The efficiency–accuracy trade-off analysis reveals significant advantages over non-YOLO architectures. While EfficientDet achieves marginally higher accuracy (mAP@[0.5:0.95] = 0.491 vs. 0.473), it requires 55.322M parameters (29.7 × our model) and 198G FLOPs (40.4 × our model). Similarly, RetinaNet demands 18.339M parameters (9.9 × our model) and 98.565G FLOPs (20.1 × our model) for comparable accuracy. This computational efficiency is critical for disaster-response applications where resources are constrained and real-time processing is essential.

4.4.2. Comparison Within YOLO Series

Comprehensive comparative analysis within the YOLO family demonstrates that LightSeek-YOLO achieves substantial model compression while preserving detection accuracy comparable to the SOTA variants YOLOv8n and YOLOv11n. Specifically, when benchmarked against YOLOv8n, our model maintains identical mAP@[0.5:0.95] performance (0.473) while achieving significant computational efficiency gains: 38.2% parameter reduction (3.01M → 1.86M), 39.5% FLOP reduction (8.1G → 4.9G), and a 36.5% model size decrease (6.11M → 3.88M). Similarly, compared to YOLOv11n, our method delivers compa-

rable detection accuracy (0.473 vs. 0.481) with remarkable resource optimization—27.8% fewer parameters (1.86M vs. 2.58M), 22.2% fewer FLOPs (4.9G vs. 6.3G) and 25.8% smaller model size (3.88M vs. 5.23M). These computational reductions collectively enhance deployment feasibility on resource-constrained devices commonly utilized in disaster-response scenarios where computational resources are inherently limited. The empirical results demonstrate that our architectural modifications to YOLOv11 effectively preserve essential detection capabilities while substantially reducing computational requirements, thereby rendering the model particularly suitable for deployment in resource-limited disaster environments where both accuracy and efficiency are paramount. The comparison of YOLOv8n, YOLOv11n and LightSeek-YOLO in disaster scenarios is shown in Figure 7.

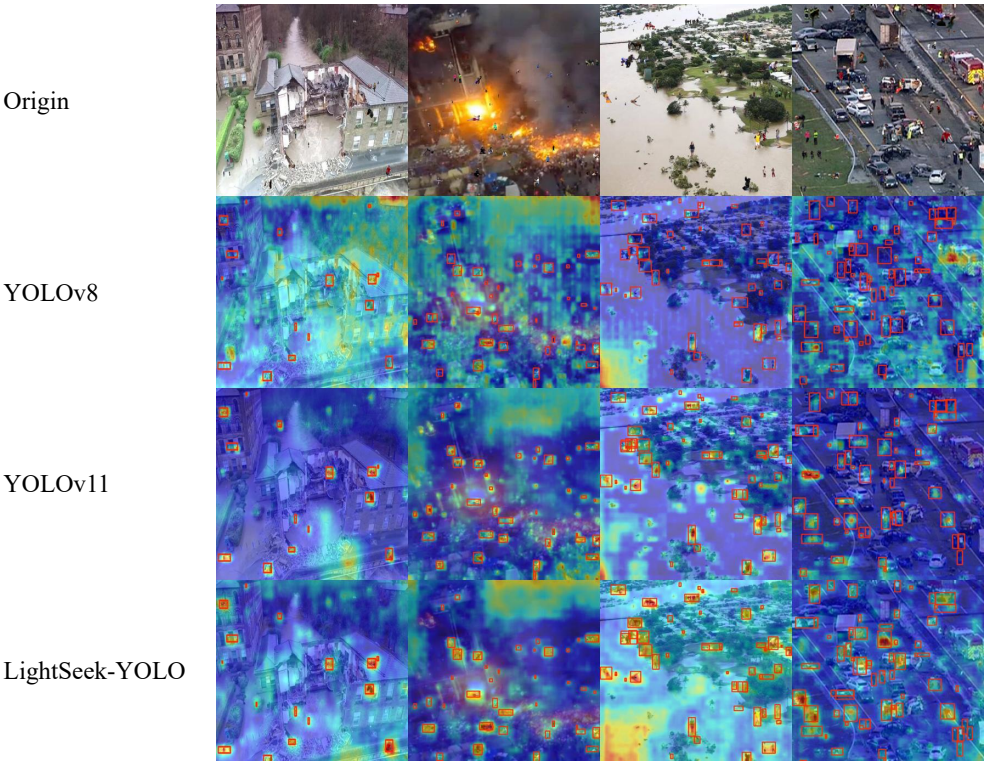


Figure 7. Comparative visualization of YOLOv8n, YOLOv11n and LightSeek-YOLO in disaster scenarios.

4.4.3. Performance Trade-Off Analysis

Our experimental evaluation highlights clear performance distinctions from existing approaches. Among non-YOLO models, EfficientDet achieves the highest accuracy but at the cost of substantially greater computational resources, which limits its practicality in constrained environments. Faster R-CNN and SSD, though widely adopted, deliver consistently lower accuracy across all metrics while still demanding more computation than our method. RetinaNet offers a moderate balance between accuracy and efficiency but remains less competitive in terms of parameter count and inference cost. Within the YOLO family, our method achieves detection accuracy comparable to YOLOv8n and YOLOv11n while substantially reducing computational overhead. This reduction is particularly advantageous in disaster scenarios, where rapid detection and deployment on resource-limited devices are essential for a timely response.

A key trade-off is the slightly lower mAP@[0.5:0.95] relative to EfficientDet (0.473 vs. 0.491). However, this 1.8% accuracy gap is offset by dramatic efficiency gains—a 96.6% reduction in parameters and a 97.5% reduction in FLOPs. In disaster-response settings, where devices such as drones or surveillance cameras operate under strict power and memory constraints, these computational savings enable broader deployment, lower

energy consumption, and longer operational lifetimes. Overall, our YOLOv11-based lightweight model strikes an effective balance between accuracy and efficiency, making it highly suitable for real-world disaster-response operations that demand both reliable detection and low computational cost.

4.5. Component Comparison

To evaluate the robustness of various architectural components within our network, we conducted comprehensive comparisons across different detection head designs.

As demonstrated in Table 2, our proposed detection head exhibits remarkable performance advantages across comprehensive metrics, achieving optimal balance between accuracy and efficiency. Our method attains an mAP@[0.5:0.95] of 0.491, representing a 2.3% improvement over the second-best model, Aux (0.480), while reaching AP@0.75 of 0.521—surpassing competing methods by 1.96–2.56% (Aux: 0.511, AttHead: 0.508). For small-object detection, our approach achieves AP@small of 0.497, demonstrating 2.26–2.90% improvement over alternatives, validating specialized optimization for minute target identification crucial in disaster scenarios. Regarding computational efficiency, our detection head maintains competitive inference speed at 617.94 FPS, nearly matching EfficientHead (618.15 FPS), while achieving superior computational efficiency with 5.6 GFLOPS—representing 11.1% and 13.8% reductions compared to Aux (6.3 GFLOPS) and AttHead (6.5 GFLOPS), respectively. With 2.42M parameters, our method requires only 4.7% more than EfficientHead (2.31M) while delivering substantially superior accuracy, demonstrating Pareto optimization of accuracy–efficiency trade-offs. In disaster contexts, these performance characteristics prove particularly valuable: the robust AP@0.5 (0.760) and AP@0.75 (0.521) enable precise localization of highly occluded targets such as limbs buried under debris, while superior AP@small (0.497) facilitates detection of minute human body fragments—often the only visible indicators of survivors in complex disaster environments—thereby directly enhancing the success rates of rescue operations and operational resilience against dynamic interference, including smoke and dust.

Table 2. Performance comparison of different detection head designs.

Model	mAP@ [0.5:0.95]	AP@0.5	AP@0.75	AP@small	FPS	GFLOPS	Param
EfficientHead	0.476	0.746	0.506	0.481	618.15	5.1	2,312,139
Aux	0.48	0.749	0.511	0.486	643.34	6.3	2,582,347
AttHead	0.478	0.747	0.508	0.483	558.17	6.5	2,596,011
Seek-DH	0.491	0.76	0.521	0.497	617.94	5.6	2,420,492

Table 3 demonstrates that our model achieves an optimal speed–efficiency balance across key performance metrics. Specifically, our method attains 617.94 FPS, representing a 7.3% improvement over SterNet (576.02), while maintaining computational efficiency with 5.6 GFLOPS—significantly outperforming FasterNet (9.2) and EfficientViT (7.9). The parameter count of 2.42M constitutes merely 18.5% of timm's requirements (13.05M), highlighting substantial model compression. Regarding detection accuracy, our model exhibits competitive performance with AP@0.5 (0.72) and AP@small (0.451) surpassing SterNet (0.719/0.445), though marginally trailing EfficientViT in AP@0.5 (0.72 vs. 0.722). The overall mAP@[0.5:0.95] (0.446) remains lower than timm (0.483), indicating an inherent trade-off between computational efficiency and comprehensive accuracy. While timm achieves superior precision (AP@0.75: 0.514), its substantial computational demands (33.6 GFLOPS) and reduced inference speed (457.81 FPS) may not satisfy the real-time requirements critical for disaster scenarios. Conversely, our model maintains >600 FPS while delivering 1.34% higher AP@small than SterNet (0.451 vs. 0.445), enabling effective detection of

minute targets within debris. For disaster rescue operations where real-time performance is paramount for victims' safety, HGNetV2 provides decisive speed advantages that facilitate rapid victim identification and localization, thereby increasing rescue success probability and mitigating risks in time-critical situations.

Table 3. Performance comparison of different backbone networks.

Model	mAP@ [0.5:0.95]	AP@0.5	AP@0.75	AP@small	FPS	GFLOPS	Param
FasterNet	0.47	0.748	0.496	0.475	551.4	9.2	3,901,959
EfficientViT	0.444	0.722	0.465	0.444	399	7.9	3,738,051
timm	0.483	0.757	0.514	0.488	457.81	33.6	13,056,003
SterNet	0.44	0.719	0.461	0.445	576.02	5	1,942,563
HGNetV2	0.446	0.72	0.468	0.451	617.94	5.6	2,420,492

4.6. Ablation Experiment

As depicted in Table 4, our lightweight model evinces remarkable adaptability to disaster scenarios. The performance modulations under extreme optimization are deliberate design stratagems tailored to enhance computational efficiency, rather than being circumscribed by architectural constraints.

Table 4. Ablation experiment result.

Baseline	Seek-DH	HGNetV2	Seek-DS	mAP@ [0.5:0.95]	AP@0.5	AP@0.75	AP@small	FPS	GFLOPS	Param
✓	×	×	×	0.481	0.748	0.512	0.485	520.5	6.3	2,582,347
✓	✓	×	×	0.491	0.76	0.521	0.497	617.94	5.6	2,420,492
✓	✓	✓	×	0.465	0.741	0.489	0.471	572.58	5	1,981,472
✓	✓	✓	✓	0.473	0.75	0.5	0.478	571.72	4.9	1,863,808

Detailed analysis of our systematic ablation reveals each module's distinct contributions: Seek-DH contributes +0.010 mAP improvement with 161,855 parameter reduction; HGNetV2 trades −0.026 mAP for a substantial 439,020-parameter reduction, representing the primary efficiency optimization; and Seek-DS recovers +0.008 mAP while achieving additional 117,664 parameter reduction. These results demonstrate that each module adds distinct value through our synergistic design strategy, where Seek-DH provides accuracy improvement with efficiency gains, HGNetV2 delivers computational optimization at controlled accuracy cost, and Seek-DS compensates for backbone limitations while maintaining efficiency benefits.

With respect to the pivotal real-time rescue requisites, the model incorporated with the Seek-DH module is meticulously optimized to prioritize computational efficiency. It attains an inference velocity of 617.94 FPS, signifying an 18.7% enhancement relative to the baseline, and mitigates the computational expense to 5.6 GFLOPS, representing an 11.1% reduction from the baseline. This intentionally efficiency-centric design directly endows the model with real-time search-and-rescue proficiency, enabling the processing of over 600 frames per second.

Noteably, this optimization of computational efficiency does not entail a compromise of critical detection performance. A balance is assiduously sustained through design: the model conserves a high level of detection accuracy (mean Average Precision, mAP@[0.5:0.95] = 0.491; Average Precision at 0.5, AP@0.5 = 0.760) and demonstrates excellence in small-object detection (Average Precision for small objects, AP@small = 0.497). This design-propelled equilibrium renders the model apt for expeditiously discerning minuscule human body components amidst rubble. Such a capacity is of paramount significance for

aiding rescuers in promptly pinpointing trapped victims within intricate disaster settings, potentially augmenting the success rate of rescue endeavors.

The model demonstrates intentional efficiency–accuracy trade-offs under extreme compression scenarios. The performance trade-off analysis of the LightSeek-YOLO module combination is shown in Figure 8. When incorporating HGNetV2, parameters reduce significantly to 1.98M (23.6% compression) and GFLOPS drop to 5.0, with mAP@[0.5:0.95] decreasing by 5.3% to 0.465, representing a deliberate design choice prioritizing computational efficiency. False detection rates increase by 3.2% in smoke occlusion test subsets, indicating that excessive lightweighting compromises feature expression in complex scenarios. The complete model configuration (Seek-DH + HGNetV2 + Seek-DS) achieves collaborative optimization through intentional trade-offs, controlling parameters to 1.86M (a 27.8% reduction from baseline) while maintaining >570 FPS performance, with the resulting accuracy profile (mAP@[0.5:0.95] = 0.473, AP@small = 0.478) reflecting our design priority of computational efficiency for resource-constrained deployment. These characteristics render the model particularly suitable for initial disaster search operations demanding high-speed response and minimal resource consumption, such as rapid UAV scanning, though limitations persist in high-precision localization (AP@0.75 = 0.489) and extreme compression robustness. Consequently, we recommend implementing dynamic model configuration switching based on operational phases: deploying the Seek-DH single-module solution during emergency search operations (FPS > 600, mAP = 0.491), then transitioning to high-precision models for fine-grained detection once disaster conditions stabilize and computational resources become available.

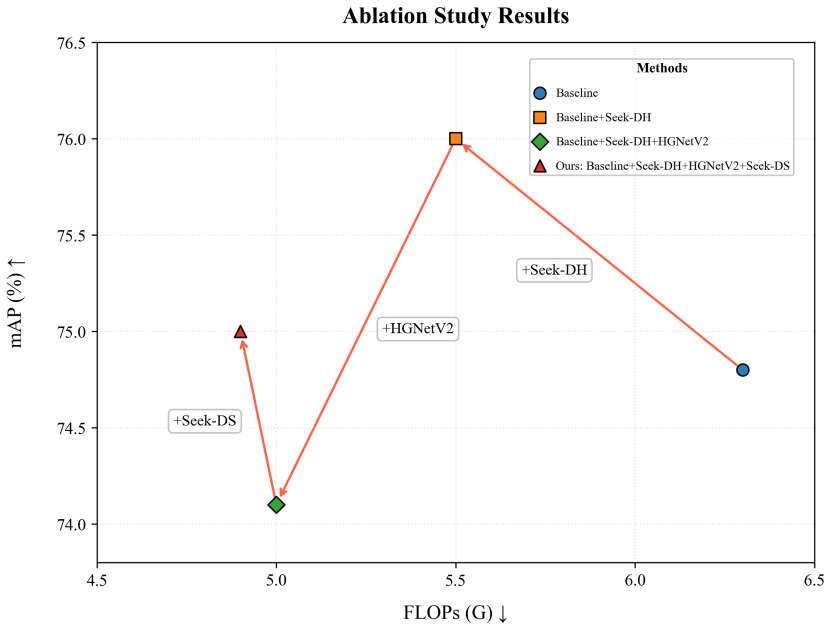


Figure 8. Performance trade-off analysis diagram of LightSeek-YOLO module combinations. The horizontal axis represents computational complexity (FLOPs), and the vertical axis represents detection accuracy (mAP). The figure shows the performance distribution of the baseline model (Baseline), followed by the model with the introduced Seek-DH detection head, which is designed to enhance feature extraction and improve detection performance, then the model with the added HGNetV2 backbone network, known for its efficient feature representation and reduced computational complexity, and finally the complete model (including Seek-DS downsampling module). The results demonstrate that through modular design, the model achieves significant lightweight effects while maintaining relatively stable detection accuracy, validating the effectiveness of the proposed method in terms of accuracy–efficiency trade-offs, which is crucial for real-world applications where limited computational resources need to be balanced with high-quality detection performance.

5. Discussion

The LightSeek-YOLO model proposed in this study demonstrates unique performance characteristics in lightweight human detection tasks in disaster scenarios. Through systematic experimental verification, we discern optimal efficiency–accuracy trade-offs. Instead of merely aiming for the maximization of accuracy, we accord precedence to the feasibility of deployment.

Module synergy effects and performance trade-offs: The ablation experiment results reveal the differential impacts of different module combinations on model performance. When only the Seek-DH module is introduced, the model achieves the best accuracy–speed balance, with the FPS improving from baseline 520.5 to 617.94 (18.7% improvement), while mAP@[0.5:0.95] increases from 0.481 to 0.491. This improvement is mainly attributed to Seek-DH's context-aware mechanism, which effectively enhances adaptability to targets of different scales through dynamic dilation rate adjustment and channel attention mechanism. Particularly in small-object detection, AP@small improves from 0.485 to 0.497, indicating that this module significantly contributes to detecting distant small-scale trapped victims in disaster scenarios.

Despite the positive impact of the Seek-DH module, further exploration of model optimization led to the addition of the HGNetV2 backbone network, which brought about different performance changes. When the HGNetV2 backbone network is added, although parameters decrease from 2.42M to 1.98M (18.1% reduction), detection accuracy declines (mAP@[0.5:0.95] from 0.491 to 0.465). This demonstrates each module's distinct value through our synergistic design strategy, where sequential ablation validates individual contributions. Although HGNetV2 effectively reduces computational complexity through depthwise separable convolution (DSC) [45], the limited cross-channel information interaction in this architecture, especially when processing complex backgrounds in disaster scenarios filled with various debris, smoke, and other interference factors, restricts the model's ability to comprehensively capture target features, ultimately leading to performance loss.

After further adding the Seek-DS module, parameters continue to decrease to 1.86M, while performance partially recovers (mAP@[0.5:0.95] of 0.473), indicating that the multi-path downsampling mechanism compensates for feature loss to some extent.

In summary, the module synergy effects and performance trade-offs observed in these experiments play a crucial role in the design and practical deployment of the LightSeek-YOLO model. Understanding these relationships enables us to make informed decisions when tailoring the model to specific disaster-scenario requirements, balancing computational efficiency with detection accuracy for optimal performance.

5.1. Comparative Analysis with Existing Methods

In comparative experiments on the COCO dataset, LightSeek-YOLO demonstrates excellent efficiency advantages. Compared to Faster R-CNN, a widely used two-stage detector known for its high-precision object detection in many applications, our method improves mAP@[0.5:0.95] by 16.8% (0.473 vs. 0.405) while consuming merely 3.1% of its parameters (1.86M vs. 60.34M). Compared to single-stage detectors, LightSeek-YOLO achieves significant model compression while maintaining comparable detection accuracy. For example, compared to EfficientDet, a popular single-stage detector renowned for its high-accuracy object detection, although mAP@[0.5:0.95] is slightly lower by 3.7% (0.473 vs. 0.491), parameters are reduced by 96.6% (1.86M vs. 55.32M) and computational cost is reduced by 97.5%. This significant reduction in parameters and computational cost in LightSeek-YOLO makes it more suitable for disaster rescue scenarios where resources are

often scarce, enabling real-time detection to be achieved by devices with limited computing power without sacrificing too much detection accuracy.

In comparison within the YOLO series, LightSeek-YOLO achieves 27.8% parameter reduction and 22.2% computational cost reduction compared to the YOLOv11n baseline while maintaining similar detection accuracy. YOLOv11n is a well-known model in the YOLO family with certain performance characteristics. This result validates the effectiveness of our proposed lightweight improvement strategy, which is particularly significant for resource-constrained disaster rescue scenarios. In such scenarios, the reduced resource requirements of LightSeek-YOLO can ensure smooth operation on various resource-limited devices, such as drones or portable detection units, facilitating timely and effective rescue operations.

5.2. Selection Strategy for Detection Head and Backbone Networks

Component comparison experiments provide important insights for module selection. In detection head comparison, Seek-DH outperforms other detection head architectures with a mAP@[0.5:0.95] of 0.491 while maintaining a high-speed inference capability of 617.94 FPS. This high AP@small value of Seek-DH indicates its strong ability to accurately detect small objects, which is of great significance in disaster scenarios where small parts of victims or important small-scale objects may be crucial for rescue operations. Particularly noteworthy is that Seek-DH achieves 0.497 in AP@small metrics, 2.3% higher than the second-best Aux detection head.

Backbone network selection presents more complex trade-offs. The timm backbone network, which is widely used in various computer vision tasks for its powerful feature extraction ability, achieves the highest detection accuracy (mAP@[0.5:0.95] of 0.483). However, the computational cost of 33.6 GFLOPS and 13.05M parameters severely limits real-time applications. In contrast, HGNetV2, designed with a focus on real-time performance in resource-constrained environments, exhibits a slightly lower detection accuracy (0.446). Yet it better meets real-time requirements with an inference speed of 617.94 FPS and 2.42M parameters, making it more suitable for rapid deployment needs at disaster sites. This trade-off between accuracy and real-time performance in backbone network selection is crucial for disaster-site applications. While high accuracy is desirable, the ability to rapidly process data and provide timely detection results is often more critical in the chaotic and time-sensitive environment of a disaster site. Therefore, HGNetV2's characteristics make it a more practical choice for such scenarios.

5.3. Applicability and Limitations of C2A Dataset

This study adopts the C2A synthetic dataset for training and evaluation. This dataset contains over 360,000 annotated instances, covering five human postures and four disaster scenarios. The dataset's distribution characteristics, with 47% of instances smaller than 10 pixels and 52% in the 10–50 pixel range, are consistent with the target scale distribution in actual disaster scenarios. However, the use of synthetic data also introduces potential limitations. In terms of texture authenticity, synthetic textures may lack the fine-details and irregularities present in real-world objects, which could potentially lead to misclassification or reduced detection accuracy for certain types of targets. Regarding lighting condition diversity, the synthetic data may not fully capture the complex and variable lighting conditions in actual disaster scenarios, such as harsh sunlight, low light at night, or uneven illumination due to debris, thus affecting the model's generalization ability.

Based on the experimental results, during the emergency search-and-rescue period after disasters (0–72 h), when rapid scanning over large areas is of utmost importance, the Seek-DH single-module configuration shows its superiority. Its high inference speed of 617.94 FPS can effectively support rapid large-area UAV scanning, while maintaining

a relatively high mAP@[0.5:0.95] of 0.491, which is sufficient to identify obvious signs of life. This balance between speed and accuracy is crucial for quickly covering large disaster areas and detecting potential survivors in the initial stage of the rescue operations. In subsequent precise localization phases, the complete model configuration can be considered. Although it reduces the speed to 571.72 FPS, the lightweight parameters of 1.86M still suit long-term deployment. This is because, as demonstrated in the experiments, in the later stages of the rescue operations when more accurate localization of survivors is required, the complete model configuration can provide relatively higher accuracy with an acceptable trade-off in speed, ensuring long-term and stable operation in the complex disaster environment.

6. Limitations and Future Work

Our evaluation was conducted exclusively on desktop hardware (RTX 4090D GPU), which does not represent the performance characteristics of UAV or disaster robotics edge devices. While our model achieves significant computational efficiency improvements (38.2% fewer parameters and 39.5% fewer FLOPs than YOLOv8n), comprehensive edge device validation remains necessary for real-world deployment confirmation. Future work will involve extensive testing on representative edge platforms to validate actual deployment feasibility in resource-constrained disaster scenarios.

7. Conclusions

Real-time object detection on computationally constrained devices in disaster scenarios presents a critical challenge for search-and-rescue operations. To address this, we introduce LightSeek-YOLO, a novel lightweight framework that strikes an optimal balance between detection accuracy and computational efficiency for victim detection in complex environments. Our approach integrates three synergistic innovations into the YOLOv11 framework: (1) the HGNetV2 backbone, which employs hardware-aware depthwise separable convolutions for efficient feature extraction; (2) the Seek-DS downsampling module, which uses a dual-branch architecture to preserve critical feature extrema and spatial patterns while minimizing information loss; and (3) the Seek-DH detection head, which leverages a unified processing pipeline with parameter sharing to reduce computational redundancy and enhance adaptability across varying target dimensions.

Comprehensive experimental evaluation demonstrates substantial performance improvements. On the COCO dataset, LightSeek-YOLO achieves a mAP@[0.5:0.95] of 0.473, matching YOLOv8n and approaching YOLOv11n (0.481), while requiring only 1.86M parameters and 4.9 GFLOPs, representing reductions of 28% and 22% compared to YOLOv11n, respectively. This results in an impressive 571.72 FPS, with computational efficiency improvements that suggest deployment potential on edge platforms, requiring comprehensive edge device validation. However, several limitations should be noted: validation is primarily conducted on synthetic C2A data, which may limit real-world generalization due to simplified lighting models and texture rendering; the architecture currently lacks integration of data (e.g., thermal or depth), which could improve performance in degraded conditions; detection of severely occluded targets (>70%) remains challenging; and validation primarily achieved using high-end GPU hardware may not fully reflect edge device performance constraints in actual disaster scenarios. Despite these limitations, LightSeek-YOLO offers a significant leap forward by overcoming the traditional accuracy–efficiency trade-off, providing a promising solution for deployment on resource-constrained hardware in emergency response. Future work will focus on incorporating real-world disaster data, developing dynamic adaptation mechanisms, and integrating fusion to enhance robustness in diverse rescue scenarios.

Author Contributions: Conceptualization, X.T. and Y.Z.; Methodology, Y.Z.; Software, Y.C.; Validation, Y.Z. and L.H.; Formal analysis, Y.Z.; Investigation, L.H.; Resources, X.T. and R.B.; Data curation, S.W.; Writing—original draft, Y.Z.; Writing—review & editing, W.S.; Visualization, Y.C.; Supervision, X.T., R.B. and W.S; Project administration, Y.Z and S.W.; Funding acquisition, X.T. and R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Foundation of Education Bureau of Hunan Province, China, the Hunan Students' Innovation and Entrepreneurship Training Program. The Hunan Students' Innovation and Entrepreneurship Training Program: Grant No. 24B0488, Grant No. S202410531096X, Grant No. 202510531018.

Data Availability Statement: The code used in this work will be made available at https://github.com/XIAOZHENGAI/LightSeek-YOLO (accessed on 25 August 2025).

Acknowledgments: The authors acknowledge Jishou University.

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

References

1. Delforge, D.A.F.; Wathelet, V.; Below, R.; Sofia, C.L.; Tonnelier, M.; van Loenhout, J.; Speybroeck, N. EM-DAT: The Emergency Events Database. *Int. J. Disaster Risk Reduct.* **2025**, *124*, 105509. [CrossRef]
2. AlAli, Z.T.; Alabady, S.A. A survey of disaster management and SAR operations using sensors and supporting techniques. *Int. J. Disaster Risk Reduct.* **2022**, *82*, 103295. [CrossRef]
3. Dong, J.; Ota, K.; Dong, M. UAV-Based Real-Time Survivor Detection System in Post-Disaster Search and Rescue Operations. *IEEE J. Miniaturization Air Space Syst.* **2021**, *2*, 209–219. [CrossRef]
4. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. *arXiv* **2017**, arXiv:1608.08710. . [CrossRef]
5. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for Thin Deep Nets. *arXiv* **2015**, arXiv:1412.6550. [CrossRef]
6. Ma, X.; Zhang, Y.; Zhang, W.; Zhou, H.; Yu, H. SDWBF Algorithm: A Novel Pedestrian Detection Algorithm in the Aerial Scene. *Drones* **2022**, *6*, 76. [CrossRef]
7. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
8. Hui, T.W.; Loy, C.C. LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 169–184.
9. Jocher, G. YOLOv5 by Ultralytics. 2020. Available online: https://zenodo.org/records/7347926 (accessed on 5 October 2025).
10. Hao, S.; Zhao, Q.; Ma, X.; Wu, Y.; Gao, S.; Yang, C.; He, T. YOLO-MSFR: Real-time natural disaster victim detection based on improved YOLOv5 network. *J. Real-Time Image Process.* **2024**, *21*, 7. [CrossRef]
11. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516. [CrossRef]
12. Chen, Y.; Li, Y.; Zheng, W.; Wan, X. Transformer fusion-based scale-aware attention network for multispectral victim detection. *Complex Intell. Syst.* **2024**, *10*, 6619–6632. [CrossRef]
13. Xu, S.; Chen, X.; Li, H.; Liu, T.; Chen, Z.; Gao, H.; Zhang, Y. Airborne Small Target Detection Method Based on Multimodal and Adaptive Feature Fusion. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5637215. [CrossRef]
14. Raja, G.; Manoharan, A.; Siljak, H. UGEN: UAV and GAN-Aided Ensemble Network for Post-Disaster Survivor Detection Through ORAN. *IEEE Trans. Veh. Technol.* **2024**, *73*, 9296–9305. [CrossRef]
15. Hou, X.; Bai, Y.; Xie, Y.; Zhang, Y.; Fu, L.; Li, Y.; Shang, C.; Shen, Q. Self-supervised multimodal change detection based on difference contrast learning for remote sensing imagery. *Pattern Recognit.* **2025**, *159*, 111148. [CrossRef]
16. Zhu, J.; Chen, Y.; Wang, L. Source-Free Cross-Modal Knowledge Transfer by Unleashing the Potential of Task-Irrelevant Data. *IEEE Trans. Image Process.* **2025**, *34*, 2840–2852. [CrossRef] [PubMed]
17. Zhu, Q.; Ma, K.; Wang, Z.; Shi, P. YOLOv7-CSAW for maritime target detection. *Front. Neurorobot.* **2023**, *17*, 1210470. [CrossRef] [PubMed]
18. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: https://github.com/Pertical/YOLOv8 (accessed on 20 July 2025).

19. Qin, X.; Li, N.; Weng, C.; Su, D.; Li, M. Simple Attention Module Based Speaker Verification with Iterative Noisy Label Detection. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6722–6726. [CrossRef]
20. Li, X.; Zhang, W.; Liu, L.; Wang, P. Two-Tier Submodel Partition Framework for Enhancing UAV Swarm Robustness in Forest Fire Detection. *IEEE Trans. Mob. Comput.* **2025**, 1–15. [CrossRef]
21. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
22. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
23. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
24. Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLOv11. 2025. Available online: https://github.com/ultralytics/ultralytics (accessed on 6 August 2025).
25. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
26. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
27. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6105–6114.
28. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
30. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Grenoble, France, 19–23 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 443–459.
31. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
32. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
33. Zhu, L.; Deng, Z.; Hu, X.; Fu, C.W.; Xu, X.; Qin, J.; Heng, P.A. Bidirectional Feature Pyramid Network with Recurrent Attention Residual Modules for Shadow Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
34. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430. [CrossRef]
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
36. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, Lille, France, 7–9 July 2015; Volume 37, pp. 448–456.
37. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef]
38. Nihal, R.A.; Yen, B.; Itoyama, K.; Nakadai, K. UAV-Enhanced Combination to Application: Comprehensive analysis and benchmarking of a human detection dataset for disaster scenarios. In Proceedings of the International Conference on Pattern Recognition, Kolkata, India, 1–5 December 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 145–162.
39. Kyrkou, C.; Theocharides, T. Deep-Learning-Based Aerial Image Classification for Emergency Response Applications Using Unmanned Aerial Vehicles. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 517–525. [CrossRef]
40. Johnson, S.; Everingham, M. Clustered pose and nonlinear appearance models for human pose estimation. In Proceedings of the British Machine Vision Conference BMVC, Aberystwyth, UK, 31 August–3 September 2010; Volume 2, p. 5.
41. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
42. Qin, X.; Zhang, Z.; Huang, C.; Dehghan, M.; Zaiane, O.R.; Jagersand, M. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognit.* **2020**, *106*, 107404. [CrossRef]

43. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
44. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
45. Sifre, L.; Mallat, S. Rigid-Motion Scattering for Image Classification. Ph.D. Thesis, Ecole Polytechnique, Palaiseau, France, 2014.

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.