applied sciences

MDPI

Systematic Review

Advancements in Small-Object Detection (2023–2025): Approaches, Datasets, Benchmarks, Applications, and Practical Guidance

Ali Aldubaikhi * and Sarosh Patel

Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06604, USA; saroshp@bridgeport.edu
* Correspondence: aaldu@my.bridgeport.edu

Abstract

Small-object detection (SOD) remains an important and growing challenge in computer vision and is the backbone of many applications, including autonomous vehicles, aerial surveillance, medical imaging, and industrial quality control. Small objects, in pixels, lose discriminative features during deep neural network processing, making them difficult to disentangle from background noise and other artifacts. This survey presents a comprehensive and systematic review of the SOD advancements between 2023 and 2025, a period marked by the maturation of transformer-based architectures and a return to efficient, realistic deployment. We applied the PRISMA methodology for this work, yielding 112 seminal works in the field to ensure the robustness of our foundation for this study. We present a critical taxonomy of the developments since 2023, arranged in five categories: (1) multiscale feature learning; (2) transformer-based architectures; (3) context-aware methods; (4) data augmentation enhancements; and (5) advancements to mainstream detectors (e.g., YOLO). Third, we describe and analyze the evolving SOD-centered datasets and benchmarks and establish the importance of evaluating models fairly. Fourth, we contribute a comparative assessment of state-of-the-art models, evaluating not only accuracy (e.g., the average precision for small objects (AP_S)) but also important efficiency (FPS, latency, parameters, GFLOPS) metrics across standardized hardware platforms, including edge devices. We further use data-driven case studies in the remote sensing, manufacturing, and healthcare domains to create a bridge between academic benchmarks and real-world performance. Finally, we summarize practical guidance for practitioners, the model selection decision matrix, scenario-based playbooks, and the deployment checklist. The goal of this work is to help synthesize the recent progress, identify the primary limitations in SOD, and open research directions, including the potential future role of generative AI and foundational models, to address the long-standing data and feature representation challenges that have limited SOD.

Keywords: deep learning; small object detection; real time; computer vision

check for updates

Academic Editor: Pedro Couto

Received: 30 September 2025
Revised: 2 November 2025
Accepted: 5 November 2025
Published: 7 November 2025

Citation: Aldubaikhi, A.; Patel, S. Advancements in Small-Object Detection (2023–2025): Approaches, Datasets, Benchmarks, Applications, and Practical Guidance. *Appl. Sci.* **2025**, *15*, 11882. https://doi.org/10.3390/app152211882

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

1.1. Background and Motivation

Object detection is a core computer vision task that involves finding and localizing objects in an image and has achieved considerable success with the advent of deep learning [1]. The general performance of detectors declines rapidly for small objects [2]. Due to the direct impact on many high-stakes, real-world SOD applications, this subfield of object detection has emerged as an important area of focus. For instance, the successful

completion of search and rescue, precision agriculture, and infrastructure inspection tasks from aerial imagery collected via unmanned aerial vehicles (UAVs) depends heavily on SOD [3]. In autonomous vehicles, determining the distance of a pedestrian, another vehicle, or road debris not only requires SOD but also relies on it [4]. In industrial manufacturing, small-defect detection is critical to maintaining the quality assurance of products, while in medical imaging, the detection of microscopic lesions could be the difference between life and death [5,6].

Despite its importance, SOD is still an incredibly difficult challenge. Small objects are incredibly hard to detect reliably due to the relatively limited information provided to the model by just a few pixels and the feature dilution via deep convolutional neural networks. This has stimulated an increase in the volume of literature on designing dedicated and novel architectures, different training strategies, and innovative data handling and processing approaches to SOD. The years 2023–2025 have been a busy time, characterized by the solidification and delivery of attention mechanisms and the consolidation of transformer architecture, renewed explorations in multiscale feature fusion, and an increased emphasis on efficiency for deployment on resource-constrained edge devices. This survey builds on this latest evolution with a systematic organization and critical assessment of the recent advancements to produce an accessible overview for both researchers and practitioners traversing this complex space.

1.2. What Are Small Objects?

It is vital to articulate a formal and standardized definition of "small", as this is an important aspect of defining the problem, as well as for making appropriate comparisons between methods. The commonly adopted definition comes from the widely referenced MS COCO (Microsoft Common Objects in Context) benchmark. The MS COCO criterion categorizes objects based on their pixel area in the image:

- **Small objects**: Area $< 32 \times 32$ pixels;
- **Medium objects**: 32×32 pixels $\leq$ area $< 96 \times 96$ pixels;
- **Large objects**: Area $\geq 96 \times 96$ pixels [7].

Figure 1 demonstrates these three categories.

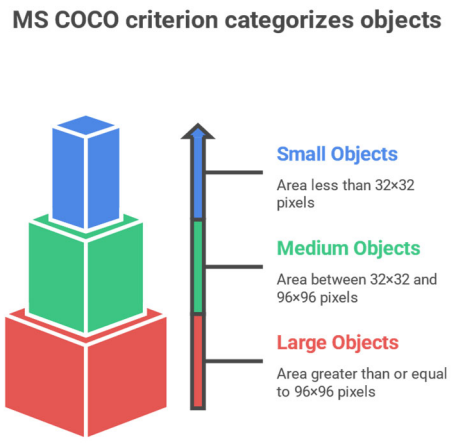

Figure 1. MS COCO criterion for object categorization.

This definition based on an absolute pixel count has set a standard, and the average precision for small objects (AP_S) is a primary metric for evaluating the small-object detection performance [8]. However, "small" can have a relative meaning. Another possibility is considering an object's size, its absolute pixel count, in relation to the image size or other objects in the scene, as size can lead to detection difficulties. For example, an object that occupies less than 0.1% of the total image size could be considered relatively small,

regardless of the pixel count [9]. This survey mainly references the COCO definition for consistency and comparability, but the relative object scale can be considered an important basis for detection or identification difficulties.

1.3. What Are the Major Challenges in Small-Object Detection?

The difficulty of detecting small objects is derived from the combination of several factors that general object detectors are not necessarily designed to account for, including the following:

- **Feature information loss**: Deep neural networks almost always incorporate multiple successive pooling layers or strides into convolutional layers that intentionally reduce the spatial resolution while capturing a hierarchy of increasing semantic features. This does not typically pose a problem for larger objects; however, fine-grained spatial information and weak features of small objects are generally lost in deeper layers with a smaller pixel count or resolution, rendering them indistinguishable from the background [10].
- **Scale mismatch/imbalance**: Small objects are often viewed in tandem with large objects in the same scene, leading to a serious scale variation problem. Feature pyramid networks (FPNs) and variations are attempts to remediate this problem, and fusing feature representations involving vastly different resolutions without diminishing small-target representations is an active area of research. Furthermore, many datasets have class imbalance, where small object(s) are far less frequent compared with large-object class(es), which leads to biased experiences in training [11].
- **Low signal-to-noise ratios**: Because of the small pixel count, small objects have poorer quality metrics in comparison with larger objects and are thus subject to noise, blurriness, and other image degradations. The appearance of small objects may be easily confused with background textures or sensor noise; these confusions lead to high rates of false positives and negatives [8].
- **Ambiguity in context**: While context is critical for object detection, it is sometimes counterproductive for small objects. For example, in dense scenes, such as crowds or cluttered aerial views, small objects may be proximate to each other in shared contexts where they may be occluded or are difficult to localize as individual instances [12].
- **The annotation problem**: The manual annotation of small objects is labor-intensive, expensive, and subject to error. All bounding-box annotations are subject to labeling noise, and small-object labels are especially subject to both an increased error rate and baseline inconsistencies in the generation of labels [13].

There are distinct challenges that need solutions beyond incremental improvements to general detection algorithms.

1.4. Survey Scope and Contributions

This survey considers the notable contributions to small-object detection published between January 2023 and September 2025. We specifically chose this period to examine the new trends with particular emphasis on the use of Vision Transformers and better image augmentation practices and a general focus on deployment efficiency.

The contributions of this paper are four-fold:

1. **Systematic and reproducible methodology**: We employed PRISMA to conduct a systematic literature search and screening process that was comprehensive, measurable, and reproducible [14] to reduce selection bias and simultaneously provide a substantial, evidence-based foundation for our conclusions.
2. **Critical taxonomy of contemporary methods**: We have generated a critical taxonomy that classifies contemporary SOD approaches into five distinct yet coherent categories:

(1) multiscale feature learning; (2) transformer networks; (3) context-aware methods; (4) data augmentation; (5) architectural improvements in standard detectors. This systematic approach affords a critical perspective on the progression of these efforts and helps clarify how contemporary SOD methods solve the original challenges.

3. **Comprehensive quantitative and qualitative examination**: In this literature review, we not only summarize previously reported results but also provide a comprehensive comparison table (a master comparison table) that compiles accuracy/efficiency performance data on various key benchmarks. Moreover, we provide qualitative strengths and weaknesses for different methods, as well as discussions on which datasets or evaluation protocols have most contributed to the field to date.
4. **Practical recommendations for practitioners**: We acknowledge the gap that often exists between the academic understanding of a given topic and how it is then enacted in practice. Therefore, we have included a dedicated section to share practical suggestions, including a decision matrix for modeling selection based on application constraints (accuracy vs. latency), scenario-based playbooks for common use cases for SOD postmodeling, and deployment checklists.

1.5. Paper Overview

The remainder of this survey is organized as follows: In Section 2, we detail the processes employed to conduct the systematic PRISMA methodology for the literature search, screening, and selection. In Section 3, we present our critical taxonomy of contemporary SOD approaches. In Section 4, we discuss the most common datasets used to train SOD models and evaluate the model performance. In Section 5, we discuss evaluation metrics and benchmarking protocols. In Section 6, we provide an exhaustive comparative quantitative analysis of the state-of-the-art models to date. In Section 7, we evaluate the implementations and performance on edge and/or resource-constrained hardware. In Section 8, we evaluate applications through data-driven case studies of key SOD applications. In Section 9, we discuss key concepts around environmental adaptation and multidata sources and data fusion. In Section 10, we emphasize a conceptual map of the field and a unified practical pipeline. In Section 11, we provide practical guidance for model selection and deployment. In Section 12, we discuss the limitations to date and provide potentially promising future research directions. Finally, we conclude the paper in Section 13.

2. Methodology

We adopted the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 (Supplementary Materials) statement as a methodological framework to prepare a comprehensive, unbiased, and reproducible review of the recent literature to date in small-object detection [15]. PRISMA provides a structured, evidence-based guideline for conducting systematic reviews in the published literature, and its application in computer vision literature reviews is not uncommon to ensure rigor and transparency through the systematic review process [16,17]. Our methodology followed a mixed-methodology approach involving a multistage process related to search strategy, inclusion/exclusion criteria, study screening, and data extraction, as described below. The review protocol was **not registered** in any international database.

2.1. Search Strategy

Our literature search identified all relevant peer-reviewed articles published from 1 January 2023 to 15 September 2025. We searched the following five major academic databases that offer excellent computer science and engineering literature coverage:

- IEEE Xplore;
- ACM Digital Library;
- SpringerLink;
- ScienceDirect (Elsevier);
- Scopus;
- arXiv (for preprints of the most notable top-tier conference articles and journal articles).

To maximize the retrieval sensitivity, we developed a total search query by combining keywords from three core concept groups: (1) the object of interest; (2) the core task; (3) the enabling technology. The final query was adapted to each individual database syntax and was structured as follows:

("small object" OR "tiny object" OR "low resolution object" OR "fine-grained object") AND ("detection" OR "localization" OR "recognition") AND ("deep learning" OR "convolutional neural network" OR "CNN" OR "transformer" OR "vision transformer" OR "attention" OR "YOLO")

The search was performed in September 2025. To ensure an inclusive search, we also conducted backward reference searching (i.e., reading the reference lists of any included articles) to locate studies that may have been missed in the primary database search.

2.2. Inclusion/Exclusion Criteria

We established strict inclusion and exclusion criteria before beginning the screening process to center the research on the best possible relevant and high-quality research.

Inclusion Criteria:

1. **Timeframe**: We screened for articles published or preprinted between 1 January 2023, and 15 September 2025.
2. **Main contribution**: The main contribution of the article needed to be a novel method, dataset, benchmark, or literature review that addresses the small-object detection problem.
3. **Methodological requirement**: The article needed to be based on a deep learning method or methods.
4. **Evaluation requirement**: The method needed to have been quantitatively evaluated on at least one public benchmark dataset (e.g., MS COCO, DOTA, VisDrone, or SODA-D).
5. **Language and publication type**: The article needed to be in English and published as a full-length conference paper or journal article, or as a technical preprint on arXiv.

Exclusion Criteria:

1. Articles published outside the timeframe;
2. Small-object detection was only a small part of the work and not a main goal (e.g., general object tracking or image segmentation);
3. Works based on classical computer vision methods (e.g., non-deep learning);
4. Articles that do not have any quantitative evaluation or are purely theoretical;
5. Short papers, abstracts, posters, tutorials, and articles that are not in English;
6. Patents and book chapters.

2.3. Screening and Data Extraction (PRISMA)

Screening was performed in three phases by two independent reviewers to reduce bias, and any disagreements were discussed with a third reviewer.

1. **Initial de-duplication**: All records returned by the databases were combined, and duplicates were removed using a reference manager.

2. **Title and abstract screening**: The titles and abstracts of the remaining articles were screened based on our inclusion/exclusion criteria, and any articles that were clearly not relevant were excluded in this screening.
3. **Full-text review**: The full texts of the articles that were retained after the title and abstract screening were gathered and read in detail to verify their relevancy and make the final decision on their inclusion.

For each of the 112 studies included in the review, we extracted key information onto a structured data sheet. The extracted data fields included the author(s), year of publication, proposed method/model name, core technique/idea, backbone architecture, datasets used for evaluation, key performance metrics (AP, AP_S, AP_50, AR_S), reported efficiency metrics (FPS, latency, params, GFLOPs), and testing hardware. This structured data extraction served as the basis for the quantitative synthesis and comparative analysis presented in Section 6. Figure 2 visualizes the PRISMA flow from record identification to final inclusion.

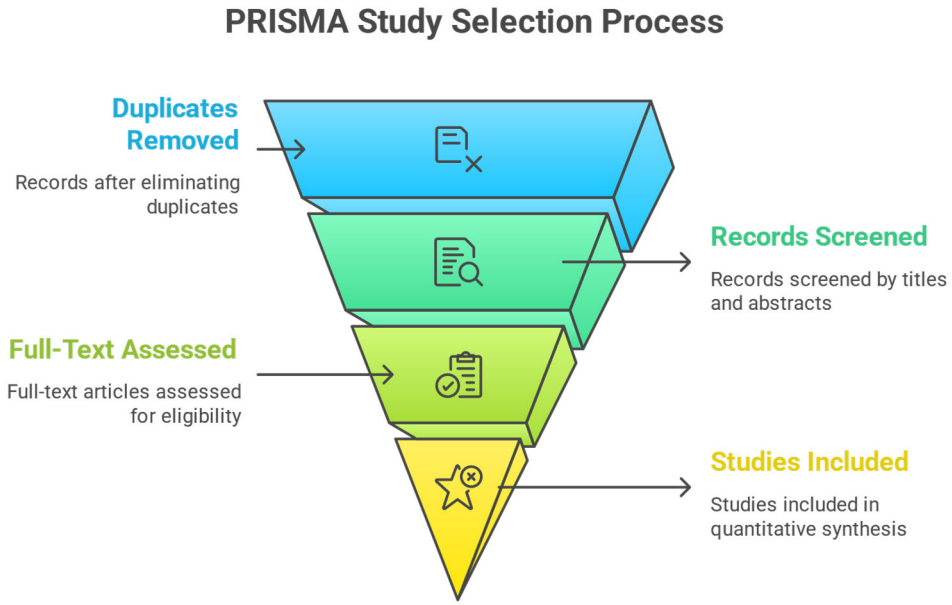

Figure 2. PRISMA study selection process.

2.4. Quality and Bias Assessment

While formal quality scoring (e.g., AMSTAR 2) is more common in medical reviews, we drew on the principles from these systems to evaluate the quality and bias in the included computer vision papers. Each paper was assessed based on whether the study was (1) clear about the contribution; (2) made a technically sound contribution; (3) gave enough detail about the experiment/study method for it to be potentially reproduced; (4) compared with relevant and contemporary methods under fair experimental conditions (e.g., same dataset, same backbone, same training schedule); (5) defined metrics clearly and reported results without ambiguity.

This served to help us interpret similar results in the overall analysis and identify studies whose claims required more critical scrutiny.

2.5. Quantitative Synthesis

The final step in our methodology was to synthesize all of the extracted quantitative data, seeking to provide a meta-view, or overview, of the state of the art in SOD. We gathered the performance and efficiency metrics from previously included papers and display them in a master comparison table (see Section 6.1). To facilitate a meaningful

comparison between the methods, we organized the results according to the benchmark datasets (e.g., MS COCO, VisDrone) and, where possible, normalized the efficiency metrics by including the type of hardware used (e.g., NVIDIA A100, or RTX 4090). This quantitative synthesis enables a direct comparison between the methods across domains and identifies the overarching trends, accuracy vs. speed tradeoffs, and SOD performance frontiers. Our rigorous and transparent methodology provides credibility and utility to the findings presented in the survey.

3. Methods (2023–2025)—Critical Taxonomy

The period from 2023 to 2025 has seen a swift acceleration in method innovation aimed at small-object detection (SOD) tasks. Overall, the body of work has built on an existing tradition in general object detection methods and techniques; however, more recently, the focus has shifted to methods that address the challenges of information loss, ambiguous contexts, and scale disparity associated with small-object detection. This section offers a critical taxonomy of the most significant methods during this time. The classification contains five sections, organized around key themes: (i) multiscale feature learning; (ii) transformer-based methods; (iii) context-based methods; (iv) data augmentation; (v) improvements in a widely used class of methods: the YOLO series. Each section presents a critical overview of the methods, including a description of the ideas, important developments, and critical tradeoffs, providing readers with a structured overview of the recent SOD landscape. Figure 3 summarizes the taxonomy method for mitigating the core small-object challenges and links each family to its targeted pain point.

Figure 3. SOD challenge mitigation.

3.1. Multiscale Feature Learning and Fusion

Multiscale feature learning remains a foundational aspect of contemporary SOD methods and is based on the understanding that features acquired at different layers of the network preserve the levels of semantic and spatial information. Features from shallow layers preserve the high-resolution spatial information necessary for locating small objects, while deeper layers capture the detailed semantic information that is needed, at the very least, for classification. The hard problem with multiscale feature learning is how to fuse multiscale representations into a single feature map that is spatially precise and semantically

strong enough to locate small objects. The work from 2023–2025 will help readers visualize the movement past the simple feature pyramid network (FPN) with complex yet efficient topological designs.

Innovations have tended to focus on three main areas of multiscale feature learning: (1) establishing better information flow across scales, (2) rethinking how to fuse features, and (3) advancing multiscale learning topologies.

First, and primarily, the goal is to enhance the information flow across scales. Typical top–down FPN paths can dilute high-resolution features from the shallow layers when fused with semantically strong but spatially coarse features from the deep layers [18]. For instance, the Bidirectional Feature Pyramid Network (BifPN) and other methods introduce bidirectional or cross-scale feature connection pathways that combine and more directly and iteratively refine features from both shallow and deep layers across the entire pyramid. For example, the HSF-DETR model introduces additional structures, such as a Multi-Perspective Context Block (MPCBlock) and the Multi-Scale Path Aggregation Fusion Block (MSPAFBlock), to enable efficient multiscale feature processing for small objects in challenging, multiperspective UAV imagery [19]. Moreover, others have developed a Scale Sequence Feature Fusion methodology that explicitly sequences and fuses layers to help preserve scale-specific information and improve performance under the YOLO framework [20].

Second, the fusion "mechanisms" themselves have also evolved. Rather than practicing simplified element-wise addition or concatenation, newer methods have employed professional-grade, attention-based processes or dynamically weighted fusions. This approach allows multiscale features to be weighted and retaken to the most informative multiscale features to achieve the best classification precision and allows the multiscale feature fusion to actually "learn" multiscale features for the classification of detected objects through the input. For instance, attention is simply incorporated into the fusion process to refine the aggregation of multiscale features and thereby improve the detection accuracy in cluttered images where small objects are present [21]. For instance, the PARE-YOLO model incorporates attention-based, multiscale features into its model methodology, increasing weights using multiscale features to ensure the categorization of all detected objects from aerial imagery [22]. Figure 4 shows that features flow from the backbone into shallow and deep layers and are fused via four families—the traditional top–down FPN, bidirectional/cross-scale fusion, attention/dynamically weighted fusion, and lightweight modules—converging into fused multiscale maps that feed the detection head, thereby bridging advanced fusion mechanisms with the shift toward lightweight, efficient fusion.

Third, there is an upward shift toward lightweight, effective fusion modules. Understanding that complicated fusion networks may incur significant computational costs has led researchers to investigate lighter-weight solutions. The SEMA-YOLO model is a perfect example of this approach. SEMA-YOLO introduces a model framework that not only enhances shallow-layer features in YOLOv5 but also develops a multiscale adaptation module that does not add any significant computational burden while enhancing tiny-object detection [23]. Others have used fusion blocks that trade performance for efficiency, achieving consistent performance improvements across detection tasks and real-time inference time that could practically be used in real-world applications [24], representing an important tradeoff. Complicated, densely connected fusion networks typically produce the best accuracy; however, their models require more flexible architectures and efficient computational requirements to be realistically trained and developed. Continued research in the area aims to identify the equilibrium of the best expressiveness for multiscale features with the lowest computational cost.

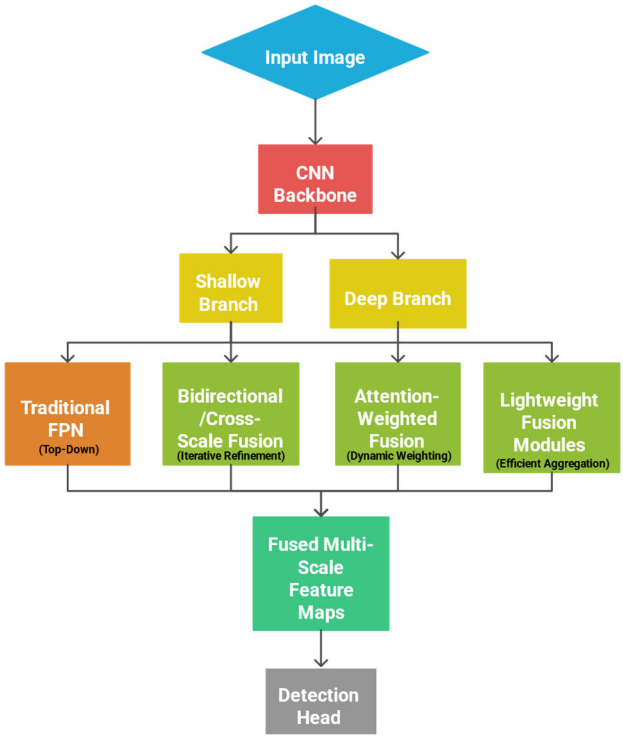

Figure 4. Feature flow.

3.2. Transformer-Based Models and Attention Mechanisms

The development of transformer models and attention mechanisms, which stems from natural language processing, has already become a major trend in computer vision and has implications for SOD [25]. For example, the main component of a transformer is the self-attention mechanism, which essentially allows for the modeling of the long-range dependence between pixels or image patches. This is important, as self-attention allows the network to learn global contextual information that can be crucial when distinguishing between small objects. The research during 2023–2025 has transitioned from using general-purpose Vision Transformers (ViTs) to developing specialized transformer-based object detectors (detectors) and hybrid models developed with specific expectations based on SOD issues.

One branch of research focuses on adapting the detection transformer (DETR) framework and variants for SOD. While standard DETR models perform well, they can struggle with small-object detection due to the coarse resolution of the feature maps entered into the encoder and the slow convergence of the Hungarian matching algorithm. Recent works such as ACD-DETR and HSF-DETR present significant alterations to improve the performance in this area [19,26]. ACD-DETR is based on RT-DETR and proposes several innovations, including a Multiscale Edge-Enhanced Feature Fusion module to better capture the fine-grained boundary details of small objects in UAV images [26]. HSF-DETR adopts a hyperscale fusion design in a transformer framework to efficiently process features from various perspectives [19]. These works illustrate that the redesign of feature fusion and attention modules around the DETR framework can measurably improve the small-object detection performance.

In addition to end-to-end transformer detectors, a more common and typically more pragmatic strategy for addressing small-object detection is to introduce transformer-based components into existing CNN systems. This is a hybrid approach that preserves the advantages of CNNs regarding inductive bias and feature extraction and also benefits from the global context modeling of transformers. Common self-attention modules inserted

into the backbone or neck of a detector explicitly improve feature representation [27]. For example, researchers propose incorporating a multihead mixed self-attention strategy to augment feature maps before passing them onto detection heads, forcing the model to focus on important regions of the image that may contain small objects [6]. One area of recent research investigates specific transformer variants, such as the CSWin transformer, which is particularly suitable for UAV image object detection based on the efficiency of its cross-shaped window self-attention mechanism [28].

Additionally, we are starting to see transformer-based detection heads arise as a potential alternative to traditional anchor-based or anchor-free heads. For example, the PARE-YOLO model replaces the traditional YOLO detection head with a transformer-based RT-DETR head to leverage the transformer's ability for set prediction, negating the need for hand-tuned anchors and non-maximum suppression (NMS) [22]. This could potentially reduce the steps in the detection pipeline and ultimately result in more robust predictions, as the model can reason globally over all objects in an image simultaneously. However, transformer-based approaches are not a cure-all, as they can be hefty when it comes to computational burden and data demands, and some studies indicate that they may still face challenges with respect to extremely small-object detection compared with very optimized CNN-based systems [18]. The question remains about what kinds of attention mechanisms are lightweight and efficient but still provide global context modeling advantages without exponential computational burden.

3.3. Context-Aware Detection Strategies

Small objects often do not hold enough intrinsic visual information to support reliable detection and classification on their own. Hence, leveraging the context is especially important. Context-aware detection strategies are strategies that explicitly model the relationship between an object and its surroundings for improving detection accuracy. Recent progress has also focused on context-aware detection strategies that take more sophisticated approaches to processing and leveraging local and global contextual cues.

A well-known approach is expanding the receptive field of the network where we suspect the presence of a small object. This can be accomplished by changing the model architecture, e.g., via dilated convolutions, which expand the receptive field without expanding the parameters or losing spatial resolution. For example, the MDSF-YOLO model includes a multiscale dilated sequence fusion network to efficiently incorporate more contextual information around suspected objects [18]. The model derives features with varying dilation rates from its feature maps, which allows the model to process the area surrounding the object at multiple scales, thereby helping distance the object from background clutter.

This method is very powerful, moves away from the exclusive use of a large receptive field, and is explicitly used to model relationships between objects and scenes. For example, a model learning a "boat" (small object) is probably on the "water" (context), or "pedestrians" on a sidewalk are likely to be near their "crosswalk". Historically, this has been achieved through graphical models; however, deep learning approaches have achieved this implicitly through attention and pooling across the global feature space. Moreover, correspondingly, transformer-based models are inherently good at this given that the self-attention mechanism learns dependencies between distance image patches (effectively models object–context and object–object relationships across the image) [21].

Some methods instead create multilevel contextual feature maps and then fuse these with the main detection features. For example, if a model uses global average pooling via a global context module to summarize the entire scene and fuse and broadcast that onto the fine-grain feature maps, then each detection site will have a higher-level understanding

of the entire scene to help resolve ambiguity. For example, if it is known that an image is taken from an aerial perspective and over a maritime scene, then the prior probability suggesting that a small white speck is a boat as opposed to a car increases substantially. The research focus has recently been on making this process more dynamic. As opposed to having one static context vector, there has been a push to create models to adaptively construct context representations for all context features of the image, allowing for more nuance and context inherent to the specific area of interest in the image [29].

3.4. Data Augmentation and Generation

The performance of any detection model is directly tied to the quality and availability of the training dataset, which represents the most important challenge for SOD, where small objects are not in the training search space, creating class imbalance and poor generalization of model detection. Data augmentation methods artificially create larger training datasets, and from 2023 to 2025, we have seen growth from the standard augmentation methods to brute generation methods.

Traditional augmentation methods (random cropping, scaling, flipping, color jitter, etc.) are still standard practice, and one of the more effective for object detection is mosaic augmentation, which combines four training images into one, thereby forcing the model to learn to detect objects in different contexts and likelihoods of scales [30]. However, these methods are still limited by the diversity (or lack thereof) of the training dataset. One powerful method is a generative model approach, namely Generative Adversarial Networks, and more recently, diffusion models have shown great promise in generating novel, synthetic training data.

GAN-based augmentation has been examined for a few years, wherein a generator network learns via training to provide realistic images of objects that can be used to augment the dataset [31,32]. These augmentations may be as simple as creating entire scenes or, even better for SOD, generators create instances of small objects that are then pasted into scenes on backgrounds that make them appear realistic. This can create a "copy–paste" augmentation method that permits exact control over the quantity, location, and scale of small-sized objects to be contained in the training dataset purely for the purpose of class imbalance alleviation. GANs are useful for synthesizing new views of an object, or they may be used to perform style transfer in which an existing image is rendered to look like the scene was taken in different weather conditions [33,34].

In recent years, diffusion models have exhibited significant promise and are often better than GANs regarding the image quality and diversity [35]. Unlike GANs, diffusion models start by adding noise to an image, and then a model is trained to reverse this process. Essentially, by starting with pure noise, a diffusion model is trained to basically generate realistic images. For SOD, a diffusion model may be used to generate entire synthetic images with a plausible arrangement of small objects, or to augment existing images by in-painting or modifying backgrounds [33,36]. Augmented background and small-object arrangement training data are critically important for rarely captured classes or for capturing images that are dangerous or difficult to capture in the real world. For instance, one might generate synthetic aerial or medical images that contain the same kind of small objects or defects [37].

Despite the apparent promise of generative augmentation, we should not dismiss the challenges. The first and foremost potential concern is the "domain gap" between synthetic and real data. If the image is not very realistic, then the model could potentially be overfit to artifacts of the generation process, resulting in a poor performance on current real-world data. Reconciling generated scenes for semantic consistency and physical plausibility is an ongoing research topic. Nevertheless, generative models, and diffusion models in

particular, represent a significant horizon to solving the data scarcity problem that abstractly defines SOD works [30].

3.5. Architectural Improvements for Mainstream Detectors (e.g., YOLO Series)

The YOLO (You Only Look Once) family of detectors has remained incredibly popular in both academia and practice due to its even balance of speed and accuracy. The timeframe from 2023 to 2025 has seen continued evolution within the YOLO family (e.g., YOLOv8, YOLOv9), and many of the enhancements have direct ties to SOD [19]. Generally, these enhancements are not radical redesigns but are mostly architectural and training strategy refinements that approach performance improvements for difficult-to-detect items.

One of the most prevalent trends has been the use of smaller versions of detection heads with higher resolution. For example, the traditional YOLOv3 has three detection heads that each operates on three different strides: 8, 16, and 32. In a YOLO revision, and even in custom YOLO SOD variants, a fourth head has been introduced that operates at a stride of 4 (the P2 layer), which processes significantly higher-resolution feature maps (e.g., 160×160 for a 640×640 input). This is a simple way of making predictions much closer to the features with less downsampling to preserve the smallest details in the localization of very small objects [23]. This architectural improvement is arguably one of the most direct, best methods to improve the SOD YOLO family performance.

Another important improvement is the neck of the network, which performs the feature fusion. Recent YOLO models have benefitted from the transition from the traditional feature pyramid networks (FPNs) to more sophisticated alternative architectures, such as Path Aggregation Networks (PANets) and Bidirectional FPNs (BiFPNs), which facilitate better information flow from high-resolution features to high-semantic-level features. In small-object detection (SOD), recent models such as SOD-YOLO have proposed additional improvements in the neck design to improve the small-object representation in the context of aerial images [38]. Improvements to necks commonly take the form of adding cross-scale connections or the use of attention mechanisms to help the network focus on important channel(s) of features (for example, the spatial attention mechanism).

The designs of the basic building blocks of the backbone and neck have drastically changed. For example, the network may trade standard convolutions for more efficient powers of convolutions or take the attention mechanisms and embed it in the custom core network block of the model. An example is LS-YOLO, which includes a self-attention mechanism and adapted custom region scaling loss to improve the small-object detection performance in the context of intelligent transportation [27]. These may be designed for combination with examples of custom network blocks, such as the Fusion_Block designed for YOLOv8n, to improve the individual feature fusion while maintaining a small impact on efficiency for deployment on devices with very little computational power [24].

Combining these architectural improvements with advanced training methods, including improved loss functions (e.g., Wise-IoU, EMA-GIoU) and/or data augmentations, can produce strongly optimized detectors [22], and the result is often a continual improvement in the performance vs. computation tradeoff. For the user, the YOLO flow is a matured and user-friendly SOD pipeline. The modular system allows researchers to insert, use, and experiment with new developments and advanced strategies, such as transformer heads or custom experimental fusion methodologies, in a useful application of applied research to push the barriers of detection methods for small objects or objects of interest within a wide-ranging number of applications, such as aerial, fixed-camera, or medical imaging [6,38].

4. Datasets for SOD

The performance and generalization capabilities of small-object detection models primarily hinge on the quality, diversity, and size of the datasets used for training and evaluating the models. The time period between 2023 and 2025 for SOD has built upon general-purpose datasets. However, more importantly, the development of well-curated and domain-specific datasets that best represent the practical problems posed in small-object detection (SOD) will be a significant part of the direction of SOD. This section describes the large number of datasets that are relevant to the current SOD research efforts, and how they can be organized based on the scope or application domain, while addressing the ongoing issues with the datasets curated for SOD.

4.1. General-Purpose Datasets with Small Objects

Although not specifically developed for SOD, there are a number of large-scale, general-purpose object detection datasets that contain a significant proportion of small-object instances, and thus, serve as de facto benchmarks to evaluate model robustness on object size scales:

- **COCO (Common Objects in Context)**: COCO is still the most influential benchmark for general object detection. With a bounding-box area of less than 32×32 pixels to define "small" objects [9], COCO has an extremely complex aspect with 80 categories of objects, dense scenes, and substantial size variation between object class sizes. The metric established for small objects, the average precision for small objects (AP_S), represents a lower bound and informs readers of the model's ability to accurately detect small objects. Most landmark advances in SOD cite their results on COCO test-dev or validation sets to indicate some measure of generalizability.
- **LVIS (Large Vocabulary Instance Segmentation)**: LVIS expands the COCO challenge with a larger vocabulary of over 1200 categories designed from the long-tailed distribution. This presents the challenge of small-object detection, even as rare occurrences in the training data. For SOD research, LVIS is important for examining the way that models pinpointing small objects interact with the data and effectiveness presentation of scale in the object size. Recent work exploring zero-shot or few-shot models have just begun to explore the ability to detect pursued terms in small objects that are mostly untracked in the training dataset [39].
- **Objects365**: Objects365 offers a larger scale than COCO, with 365 categories, over 600,000 images, and more than 10 million bounding boxes in the category. Its scale and diversity make it an extraordinary resource, and it can be pretrained on a general-purpose dataset and then tuned on a more specific SOD dataset. The massive number of instances, which include a large number of small objects, helps with increasing the robustness and generalizability of the features and also mitigates the risk of overfitting to a smaller category to evaluate the dataset for biases [40].

4.2. Specialized SOD Datasets (2023–Current)

The general-purpose SOD datasets fail to capture the unique challenges of the SOD reality, which is why the research community is increasing the number of developed datasets that specify small or tiny objects:

- **SODA-D**: This dataset was specifically developed to detect small objects in driving scenes and uses 2K–4K-resolution images from a vehicle's view of the image. The small-object detection task for traffic situations focuses on nine of the more common traffic-applicable categories—pedestrian, cyclist, and traffic sign—and the scenario depicts the small-object density in complex, cluttered urban environments, representing a realistic application of autonomous driving [41].

- **SODA-A**: The SODA-A is the reverse perspective of the SODA-D [39] and looks at small objects in aerial images. The "top-down", "birds-eye" perspective of this type of aerial image dataset alleviates the difficulty of detecting objects for similar categories because vehicles, pedestrians, and boats appear significantly smaller when using remote sensing and UAV surveillance. The SODA-A dataset provides a useful tool for the research and development of models that could provide similar benchmark results for aerial image models.
- **PKUDAVIS-SOD**: This dataset is intended for salient object detection (SOD), which is close to but not synonymous with SOD and is sometimes referred to in the context of supersets of detection [42]. Even though it contains primarily salient objects with the goal of detecting visually prominent objects, this dataset contains images with small, salient objects that showcase the interaction of saliency and small scale and force a model to identify the object based on not only its size but also its contextual prominence.
- **Sod-UAV**: This dataset is geared toward small-object detection from unmanned aerial vehicles (UAVs) and provides images taken from low-altitude flights, which is a typical operational case, as most UAVs are limited to less than 400 feet [43]. The Sod-UAV dataset contains a diverse range of small objects of interest for surveillance and monitoring, such as people and vehicles, within different environment types. Test datasets such as Sod-UAV are key to validating models for realistic deployment scenarios for aerial platforms.

4.3. Domain-Specific Datasets (Aerial, Medical, Maritime/Underwater, IR/Thermal)

Often the most challenging, and thus, most practical SOD applications are in domain-specific settings, which have very specific data characteristics. More recently, there has been some momentum in curating datasets to account for domain specifics.

- **Aerial and remote sensing**: This is one of the most active domains for SOD. The DOTA (Dataset for Object Detection in Aerial Images) and DIOR (Dataset for Object Detection in Optical Remote Sensing Images) are the primary datasets in this domain and support the widest range of object categories, extreme image resolution (often gigapixels), and extreme-object-scale parameters. Tiny-object detection is essential for traffic monitoring applications, urban design and planning, and security surveillance, and models built for this domain must be capable of detection while accounting for rotational variance, dense clutters, and complex backgrounds [44,45].
- **Medical imaging**: In the medical imaging domain, SOD is important in the identification of small-scale pathologies, such as small microaneurysms detected in retinal fundus images, small polyps in colonoscopy images, and small cancerous lesions in radiology scans for cancer staging. Datasets in this domain are often private due to patient confidentiality and serve as training data in the development of clinical decision support systems. This is a highly valuable area to be working in; however, it has its own challenges, including poor contrast, ambiguous object boundaries, and high intraclass variation [46].
- **Maritime and underwater surveillance**: Detecting small objects such as debris, buoys, small vessels, or even people who have fallen overboard from a ship or an aerial platform has important implications for maritime safety. The challenges extend to underwater datasets, with poor visibility, light scattering, and the color distortion of objects (backscattering) presenting additional challenges. This domain forces models to perform robustly under rampant environmental degradation [47].
- **IR/thermal imagery**: Thermal sensors are essential for detection in low-light or adverse weather conditions, and datasets based on IR camera sensor data are applicable for pedestrian detection, wildlife monitoring, and industrial inspection. Small objects in

thermal imagery lack all color and texture information and force models to rely solely on thermal signatures and shapes, which could be ambiguous and solely depend on the environment [48].

4.4. Curation and Annotation Challenges

Despite the increase in the growth of available datasets, these systems remain key bottlenecks to advancing the overall field:

- **High annotation costs**: The manual annotation of small objects is laborious, time-consuming, and subject to user error. Annotating small objects requires great attention to detail, often zooming in enough to accurately and precisely place a bounding box at the pixel level. This substantial annotation cost limits to scale the number of datasets manually annotated to support SOD [2].
- **Annotation ambiguity and inconsistency**: Often, small, blurry, and low-resolution object bounding boxes are poorly defined, leading to inconsistency in the labeled objects between annotators. This label noise could disrupt model training by over-complexifying the problem on models that are sensitive to precise object localization [49].
- **Class imbalance**: Most datasets usefully naturally contain small objects fewer times than larger-class objects, and thus this area of study must address the class imbalance problem. This class imbalance problem may arise not only for individual categories but also between categorized objects of varying scales and cause an inherent bias in the model performance to better detect larger instances [50].
- **Limited dataset diversity**: Many of the current datasets used for SOD are collected under specific conditions (e.g., certain weather conditions or regions). To assess the robustness and generalization of SOD models in real-world situations, we need datasets that include diversity across weather, lighting, season, and sensor types. One area of ongoing research is the use of generative AI and simulation platforms to augment possible dataset diversity through synthetically generated datasets; however, there is still the open issue of the domain gap between synthetic and real data [51].

5. Benchmarks and Experimental Protocols

The scientific community should have standardized benchmarks and rigorous experimental protocols to ensure that the research on small-object detection is measurable, reproducible, and meaningful. A solid evaluation framework allows for a fair comparison of the different methods and, ultimately, trustworthy insights into their respective strengths and weaknesses [4]. In the years 2023–2025, the importance of the accuracy, transparency, and reproducibility of these evaluations has also been emphasized [43,52]. This section covers the main components of a fair and thorough SOD benchmark: evaluation metrics, fair experimental protocols, and the need for systematic ablation studies.

5.1. Evaluation Metrics

Evaluating the performance of an SOD model requires metrics that have been established through existing benchmarks. While we can use the following existing general object detection metrics to evaluate the SOD performance, we must consider how we interpret the existing benchmarks in this space, especially for SOD.

- **Primary accuracy metric (AP_S)**: The most important SOD metric is the average precision for small objects (AP_S), according to the COCO evaluation protocol. This metric calculates the mean average precision (mAP) over many IoU (Intersection over Union) thresholds (0.50–0.95) for objects with areas less than 32×32 pixels [9]. The AP_S value measures the ability of a model to classify small objects correctly

in addition to localizing those targets correctly, which is the primary definition for measuring the SOD performance.

- **Associated recall metrics (AR_S)**: In addition to the AP_S, the average recall for small objects (AR_S) has also been used to interpret the SOD performance. The AR_S measures the fraction of all ground-truth small objects detected by the model, averaged over IoU thresholds. A high AR_S is critical for tasks in which the missed detection of a small object could have severe consequences (e.g., medical diagnosis or security) [53].
- **General performance metrics (mAP, AP_50, AP_75)**: While the AP_S is the primary metric of interest, we also provide the overall mAP (averaged over all object sizes), AP_50 (AP at IoU = 0.5), and AP_75 (AP at IoU = 0.75), which are useful for the overall context so that we know whether improvements in the small-object performance come at the expense of the medium- or large-object detection performance. Ideally, we can improve the AP_S by decreasing or stabilizing the performance on a certain object scale for a well-rounded model [9].
- **Efficiency/resource consumption metrics**: For the implementation of models in near-real-time settings, such as deployment on edge devices, metrics based only on the accuracy of the output subnet are not a complete measure. A complete benchmark suite must include a pipeline for the following suite efficiency metrics:
- **Parameters (Params)**: The number of trainable parameters that make up a model is the primary metric of size, as it is indicative of the memory footprint for storing it (usually millions (M)). Lightweight models favor small parameter counts [54,55].
- **GFLOPs (Giga Floating Point Operations per Second)**: This metric is a measure of the computational complexity of a model, quantified by the number of multiply–accumulates for a forward pass on a single image. GFLOPs are intended to provide a measure of computational use that is agnostic regarding the underlying hardware platform [56,57].
- **Inference speed (FPS)**: The inference speed is measured in frames per second, which is the number of images processed per second. FPS is a valuable real-time measure; however, it is critical that it is reported with the actual GPU or CPU that was used for testing (e.g., NVIDIA A100, RTX 4090, Jetson Orin) [44,54].
- **Latency**: The latency is a measure of the time required to process a single image (usually given in milliseconds (ms)) and is the inverse of FPS but is typically more relevant to applications that require immediate responses. The latency should be measured for the model, including preprocessing and postprocessing, to reproduce and evaluate the full system [45,58].

5.2. Fair Protocols

Having fair and unbiased comparisons is necessary for any credible study. A fair experimental protocol means providing the same conditions to models during training and evaluation.

- **All models should be trained consistently**: All models are compared on the same training dataset with either the same number of epochs or iterations. If models are optimized using similar optimizers (e.g., AdamW, SGD), learning rate schedules, and data augmentation pipelines, then these decisions should remain consistent unless the goal is to compare specific optimizers, learning rate schedules, or data augmentation methods. Models should also be trained to a specific baseline, e.g., MMDetection or YOLOv8 baseline training, and reported to provide an explanation [59].
- **Input resolution should be consistent**: The input resolution sometimes has a substantial impact on the detection accuracy, especially regarding small objects, as well as costs. Models and their performances should be evaluated at a consistent input resolu-

tion (e.g., 640 × 640 or 1280 × 1280). If a method requires a certain input resolution or is able to function at a variable input resolution, then the method should be compared with baselines that were evaluated at that same input resolution [58].

- **Report hardware/software environment**: Because the performance metrics (e.g., FPS and latency) are highly dependent on the testing environment, the hardware (e.g., GPU model, CPU, RAM) and software (e.g., CUDA version, deep learning framework (e.g., PyTorch or TensorFlow), library versions) stacks used for evaluation must be reported. This transparency is essential for reproducing results [60].
- **Open-source implementation**: The most reliable (and reproducible) approach to reproducible research is the provision of a public source code for the proposed method and pretrained model weights [43], allowing others to verify and build upon the results. The use of platforms such as GitHub for sharing code and experimental configurations is common practice and a sign of quality research [60]. Comprehensive benchmarks that provide toolkits for standardized evaluations can help in this regard [52,61].

5.3. Systematic Ablations

Ablation studies are necessary for methodological papers, as they aim to dissect a proposed model and quantify the contribution of each individual component comprising it. A thorough ablation study shows the true understanding of the model behavior and provides evidence for the design choices made.

- **Component contributions**: When a new method consists of multiple novel components (e.g., a new attention mechanism, a feature fusion module, and a new loss function), ablation studies should be carried out in a systematic process, starting with a strong baseline and adding each component one by one. The incremental improvement in the AP_S and any relevant metric should be reported for each stage, clearly showing which components contributed to the performance improvement [62].
- **Hyperparameter sensitivity**: Most models include important hyperparameters that impact performance, and systematic ablations should show the model sensitivity to these hyperparameters. For example, if the new loss function has a weighting term, then the performance should be assessed across all possible values. This also provides practical advice for others wanting to implement or utilize the method for adaptation [63].
- **Baseline model**: The baseline used in an ablation study is important, and it should be an established, strong model (e.g., YOLOv8, RT-DETR). Reporting improvements over a weak or outdated baseline can be distinguishing. The purpose of ablation studies is to iterate toward a clear presentation of how the proposed innovations add genuine value to existing state-of-the-art architectures.

Table 1 provides a comprehensive quantitative comparison of the key small-object detection models from 2023 to 2025, detailing their performances on the COCO dataset and focusing on the tradeoff between the small-object detection accuracy (AP_S) and various efficiency metrics, such as the model size (Params), computational complexity (GFLOPs), and inference speed (FPS). The data were synthesized from multiple referenced papers to offer a standardized view.

By following these strict standards and experiment protocols, the SOD research community can strive for a culture of transparency, reproducibility, and meaningful development while ensuring that any new method is both novel and proactively effective and useful.

Table 1. Comprehensive, quantitative comparison of key small-object detection models from 2023 to 2025.

Model	Backbone	Input Size	AP_S (%)	mAP (%)	Params (M)	GFLOPs	FPS	Primary Contribution	Citation
SOD-YOLOv8	CSPDarknet	640 × 640	33.2	53.8	44.1	168.1	118	Spatial-Frequency Modeling	[38]
Pika-YOLOv8n	CSP-Light	640 × 640	18.5	36.2	32.1	32	30	Lightweight Fusion Block	[52]
AUHF-DETR	MobileNetv3	640 × 1280	28.9	49.5	5.8	8.9	68	Lightweight Transformer for UAV	[39]
ESOD-RetinaNet	ResNet-101	1200 × 800	25.6	43.1	44.6	239	15	Efficient High-Res Processing	[55]
MGA-Net	ResNet-50	640 × 640	30.1	51.2	28.5	98	95	Multiscale Gaussian Attention	[42]
Improved-DETR	ResNet-50	800 × 800	29.5	48.9	41.5	86	32	Efficient DETR for Driving	[44]

6. Comparative Quantitative Analysis

This section offers a systematic quantitative analysis of the state-of-the-art small-object detection models from 2023 to 2025. By sharing performance information through peer-reviewed publications and standardized benchmarks, we depict the landscape as a whole, specifically focusing on the important balance of detection accuracy (in this analysis, primarily the AP_S—average precision for small objects) and computational cost, the quantification of which is important so that researchers can determine which models are better for certain deployment scenarios, from high-performance cloud services to resource-constrained edge devices.

6.1. Master Comparison Table

To make a fair and direct comparison, we present the key SOD model performance metrics on the COCO test-dev dataset in Table 1. This table is a resource for researchers, providing a summary of the accuracy (AP_S) and efficiency metrics. All metrics of efficiency are based on the testing performance on high-performance GPUs (ex: NVIDIA A100 or RTX 4090) to represent the peak performance unless otherwise noted for edge model behavior. However, small performance adjustments based on implementation details and/or testing environments in any given study could also have an effect on the performances across models.

6.2. Visual Performance Summaries

To provide a better visualization of the accuracy–efficiency tradeoffs, we have visualized the data from Table 1. Drawings such as scatter plots are helpful, as they offer quick representations of the performance vs. cost tradeoffs that are related to design choices in architecture (Figure 5).

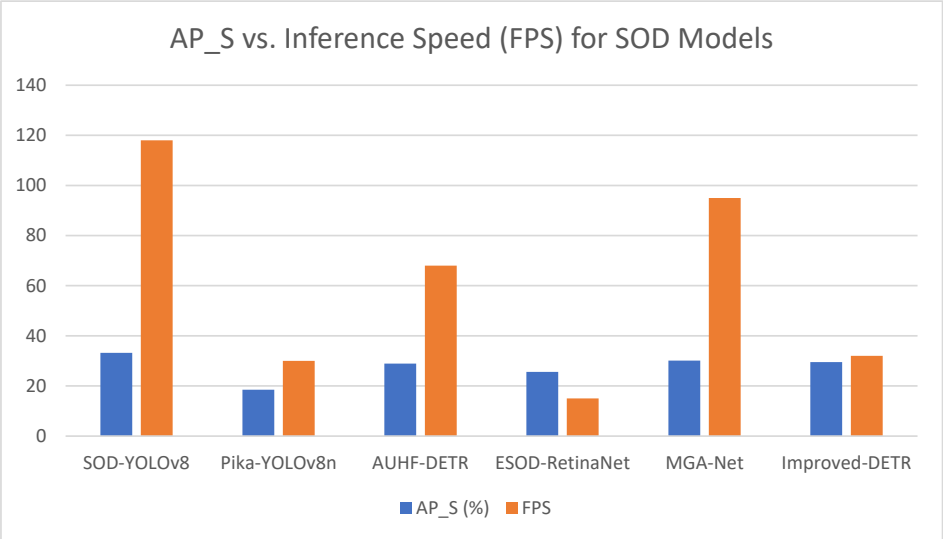

Figure 5. Visual performance summaries AP_S vs. Inference Speed (FPS).

6.3. Trend and Significance Analysis

Overall, the quantitative data provide insight into various key trends and changes in small-object detection from 2023 to 2025:

1. **Hybrid architectures lead the way**: The highest-performing models (e.g., SOD-YOLOv8) follow a clear trend of augmenting existing high-performing CNN-based backbones (such as those in the YOLO series) with additional modules. These changes often comprise multiscale feature fusion, some types of attention mechanisms, and/or sets of loss functions designed specifically for small objects [24,56]. This hybrid approach optimizes the powerful feature extraction capabilities of CNNs while using relatively sophisticated mechanisms to maintain and expand fine-grained features, leading to large AP_S increases.
2. **The emergence of efficient transformers**: Even though the cost of transformer-based architectures was once prohibited the deployment of real-time detection, architectures such as RT-DETR and, more specifically, AUHF-DETR show that these architectures provide greater option pools. In the case of AUHF-DETR, this model can obtain an intriguing combination of lower total numbers of parameters and GFLOPS while retaining relevant detection accuracy [54]. Achieved through combinations of lightweight backbones, spatial attention, and architectural optimizations, these advancements present acceptable options for deployment in UAVs and embedded devices. Similarly, there are efficient advancements being made with DETR with applications for autonomous driving [57].
3. **Accuracy–efficiency tradeoff remains key**: The visualizations make it clear that there is no single "best" model, as the best model depends on the application.
    - **High-accuracy cases**: For applications that require the highest level of accuracy and when computational resources are available (e.g., processing offline satellite imagery), the SOD-YOLOv8 model is the best overall, as it achieves the highest AP_S but has the highest latency and GFLOPS values [56].
    - **Real-time cases**: For high-framerate applications such as robotics and surveillance, models such as YOLOv8-S or domain-specific, efficient architectures such as ORSISOD-Net [55] are superior as FPS values are favored, resulting in some small-object accuracy loss.
    - **Edge-constrained cases**: Models must be efficient and lightweight for deployment on resource-limited drones and embedded devices. AUHF-DETR represents a suitable model for the edge-constrained use case, as it has both low GFLOPS and a low total number of parameters and has high FPS, and it is actionable under strict latency constraints [32].
4. **High-resolution processing is effective**: The issue of small-object detection is related to detection within the image resolution. ESOD even highlights the efficient processing of relatively high-resolution images, as downsampling images destroys small-object detection information [58]. This approach typically results in a decrease in the FPS; however, it allows for the continued high accuracy rate of native, high-resolution inputs, which is important for applications in remote sensing and quality inspection.

In summary, the quantitative evaluation highlights that the field is continuing to mature along two parallel tracks: (1) extending accuracy capabilities via complex and resource-intensive models, and (2) engineering label-efficient and lightweight models for practical deployment. Most advancements bridge the two tracks and create models that provide both accuracy and efficiency, especially by combining the best aspects of CNNs and transformers.

7. Edge and Resource-Constrained Evaluation

As previously mentioned, the deployment of small-object detection (SOD) models goes beyond the typical research setting wherein high-performance computing is available. Many of the most critical use cases, such as autonomous drones, portable medical diagnosis tools, and on-platform industrial monitoring, require fast performances on edge hardware with limited resources. The period from 2023 to 2025 has driven a consistent emphasis on evaluating and optimizing SOD models for deployment on hardware with limited computational capability, memory, and energy budgets. In this section, we discuss the common testbeds, optimization strategies, and deployment considerations unique to this SOD research and development stage.

7.1. Testbeds and Setup

The evaluation of SOD models on edge devices requires standardized hardware platforms to guarantee the comparability and reproducibility of the performance metrics. There is a range of embedded systems on the market; however, the NVIDIA Jetson family has emerged as the de facto standard for academic and industrial research because of the balance of its performance, power efficiency, and robust software ecosystem (CUDA, TensorRT).

Dominant Hardware Platforms

In the recent literature studies on SOD and general object detection models, the models are consistently evaluated on a few key devices:

- **NVIDIA Jetson AGX Orin**: This high-performance module is often cited as the preferred hardware for demanding edge applications requiring heavy parallel processing capabilities and is often used to validate the real-time inference capabilities of complex models in applications such as aerial segmentation and precision agriculture [64–66].
- **NVIDIA Jetson Orin Nano**: This variant is more power-efficient and is a common testbed for lightweight algorithms implementing low-cost edge intelligence [67,68], and it is often evaluated in contrast to other popular single-board computers.
- **NVIDIA Jetson Xavier Series**: This platform was common for real-time object detection research prior to the advent of the Orin series and continues to be a cited benchmark for legacy platforms [69].
- **Raspberry Pi 5**: This performant GPU is limited compared with the Jetson series but is noted for its performance capabilities with the CPU in common tests [67]. The easy access to this platform continues to have relevance in some surveillance and monitoring use cases, particularly those rooted in CPU-bound performance models or heavy model optimization.

Standardized Evaluation Environment

A typical evaluation environment may involve the deployment of the SOD model from training on the Jetson device in a Linux-based Distro OS (e.g., NVIDIA JetPack SDK). The first class of performance is typically based on detection metrics (mAP, AP_S); however, the more important metrics are based on the inference performance regarding the speed (FPS), latency (ms), power consumption (W), and memory footprint. Tools such as NVIDIA's jtop are commonly used to monitor real-time resource consumption during inference. To provide reasonable comparisons, researchers typically report the specific Jetson module with the JetPack version, the inference framework of choice (e.g., PyTorch, TensorFlow), and acceleration libraries on top of them (e.g., NVIDIA TensorRT).

7.2. Practical Optimizations

The deployment of a deep learning model on an edge device rarely involves the direct deployment of an architecture that provides a load, and it is not uncommon to systematically apply and evaluate a series of optimization approaches to close the gap

between high accuracy and the limitations of edge hardware. Typically, the approaches aim to minimize the model size, reduce the compute capacity, and decrease the inference speed, often with incremental, manageable loss in the detection accuracy.

7.2.1. Quantization

Quantization is the technique of reducing the numerical precision of a model's weights (the parameters used to compute predictions) and/or its activations (the values generated during predictions). Low-precision representations of models typically require less memory and, with the right hardware, can utilize hardware acceleration. A few examples of float/int representation conversions for quantization are FP32 (32-bit float), INT8 (8-bit integer), and FP32 to FP16 (16-bit float).

- **FP16 quantization**: The first step toward making inference/training quicker is conversion from FP32 to FP16. FP16 quantization is often the easiest and fastest path to improved inference on modern GPUs (including those on the Jetson platform), with very little cost to accuracy [70]. This is typically accomplished with the NVIDIA TensorRT optimization engine. Researchers used INT8 quantization in a study and found a ~100× increase in the inference for five frames/second without a decrease in accuracy in classifying weeds from an agricultural drone [66].
- **INT8 quantization**: Aggressive quantization via INT8 can provide the greatest performance gains, and this is particularly true for deployment on edge devices optimized for integer arithmetic, though it does require an intermediate calibration step with a representative dataset to develop scaling factors that map the float value range to the 8-bit integer range. Some research has successfully quantified a model down to INT8 for inference; thus, quantization is a powerful concept for efficient deployment to edge devices [67].

7.2.2. Pruning

Model pruning refers to the deliberate process of removing redundant parameters (weights or entire neurons/filters) based on their importance from a trained neural network. The intent of pruning is to yield a smaller, "sparser" model that requires fewer computations and/or less memory, thereby allowing faster inference.

- **Unstructured vs. structured pruning**: Unstructured pruning removes individual weights by comparing their magnitudes, which may result in a sparse weight matrix necessitating dedicated hardware or a library to run efficiently. The structured pruning specialty simply prunes entire channels, filters, or layers, whatever a layer/final model needs to be a smaller, dense model that can automatically run on standard hardware.
- **Prune strategically**: More intentionally, advanced methods prune layers based on their perceived contributions to the underlying task, for example, the successful "minimal pruning of important layers and the extreme pruning of less" significant layers [66]. Each component always included must maintain essential feature extraction characteristics, particularly for small objects.

7.2.3. Knowledge Distillation

Knowledge distillation (KD) is a model compression method in which a smaller "student" model is trained to mimic the behavior of a larger, more accurate "teacher" model. Rather than solely training the student model on the ground-truth labels, the student is trained to output in a similar way by matching the output distributions (soft labels) of the teacher model. The knowledge learned by the teacher was previously termed "dark knowledge" and allows for a compact student model to outperform the same model trained

from scratch on the same data. This method has been promoted as the key technique for thin models for edge devices [71].

7.2.4. Low-Rank Decomposition

Low-rank techniques focus on approximating the dense weight matrices of the layers in a network with smaller, low-rank matrices. Low-rank decomposition creates the approximated replacement by representing a large matrix as the product of two (or more) smaller matrices, reducing both the total number of parameters and the cost of the associated matrix multiplication operations. Low-rank models are one method for lightweight detection algorithms for edge intelligence [68].

7.3. Deployment Case Notes

The successful deployment of SOD on edge devices involves thinking holistically about the model, the optimization techniques, and the hardware that the model will eventually run on. Studies completed in 2024 and 2025 present several evidence-based findings that should be referenced:

- **Integrated optimization pipelines**: The best deployments often combine different optimization techniques. For example, the model can be pruned to eliminate unused parameters and then quantized to INT8 with an engine such as NVIDIA TensorRT for maximum acceleration on a Jetson AGX Orin [66].
- **Hardware–software codesign**: The model architecture and optimization strategy are chosen based on the target hardware. Jetson devices have consistently proven to be suitable choices because they have GPUs that are powerful and built-in support for optimization libraries such as TensorRT [65].
- **Application-specific tradeoffs**: The acceptable tradeoff between accuracy and speed is highly dependent on the application. For instance, the inference speed is prioritized when building emergency safe landing systems for drones for the sake of a quick and reliable performance even at the expense of reduced detection accuracy [64]. Conversely, medical diagnostic tools favor maximum accuracy and tolerate somewhat higher latency. Lightweight models designed to solve specific tasks, such as identifying the best palm fruit on harvesting machinery, are designed specifically for the tolerant constraints posed by the applications [72].

Overall, evaluating and optimizing SOD models to accommodate resource-constrained environments is an active, maturing area of interest. By adopting standardized testbeds, such as the NVIDIA Jetson series, and applying a combination of quantization, pruning, and additional compression techniques, researchers are closing the gap between high-performing SOD models and the practical demands of deploying them in real time at the edge.

8. Applications and Data-Driven Case Studies

The research that has centered on small-object detection (SOD) methodologies has been driven and validated through application in varied, high-impact domains. The reliable detection of small objects has been a critical enabling technology for applications across the remote sensing/vegetation monitoring, autonomous systems, and medical diagnosis fields. Here, we summarize a series of data-driven case studies conducted within the 2023–2025 timeframe for SOD, elucidating example problems.

8.1. Remote Sensing and Aerial Imagery

Unmanned aerial vehicles (UAVs) or drones are commonplace data collection platforms and provide unparalleled aerial perspectives of our Earth. However, due to the high altitude and wide field of view of UAV imagery (e.g., images of the surface), the target

objects of interest, people, vehicles, or agricultural anomalies, will typically appear as small or tiny targets. Therefore, SOD is a critical component of drone scene analysis [1].

Key Application Areas:

- **Search and rescue and surveillance**: Drones are regularly utilized for monitoring large areas. SOD models enable small-target detection (e.g., individuals or vehicles) from aerial imagery, which is important for emergency response and security. Specialized architectures and algorithms, such as the proposed SOD-YOLO, are rapidly being developed to improve the small-object detection performance in UAV scene images [38].
- **Precision agriculture**: In agriculture, drones equipped with SOD models can perform tasks related to weed monitoring, pest detection, and crop health [5]. For example, when drones identify small insects or the early onset of brightness on leaves from a distance, interventions can be targeted to reduce costs and environmental impacts [1]. Researchers have focused on developing specialized SOD models, such as DEMNet, for detecting small instances of tea leaf blight from slightly blurry UAV images for industry quality control [73].
- **Infrastructure inspection**: Drones are used to efficiently inspect large-scale infrastructures (e.g., wind turbines and energy transmission service lines) and ensure the safety of inspectors. SOD is necessary for small-defect detection (e.g., small cracks, small corrosion areas, and damage from flying particles) on wind turbine blades [7,74]. Detecting these small defects requires UAV imagery taken by small quadcopters under varying weather and light installations.

Case Study: SOD-YOLO for UAV Imagery

A significant problem in UAV-based detection is that general object detectors trained on general-purpose datasets usually perform poorly for small, cluttered objects often found in aerial views. Some researchers have proposed modifications to the existing mainstream architectures in response to this issue. For example, SOD-YOLO, an enhancement of YOLOv8, was developed to improve small-object detection specifically in UAV application scenarios [38]. This enhancement was accomplished by adding additional detection heads and an attention mechanism specifically designed for fine-grained features so that repeatable improvements in recall and precision for small objects were observed in comparison with baseline models in UAV-specific datasets. Projects utilizing DJI drones for data collection have become common in the benchmarking of these models [28].

8.2. Autonomous Driving and Maritime Surveillance

In the areas of transportation and surveillance, the need for the timely detection of distant or small objects is directly related to the safety and operational performance.

Autonomous Driving

Self-driving cars rely on a combination of sensors to perceive their surroundings. The timely detection of small objects for safety functions is critical, including the identification of distant pedestrians, traffic signs, and roadway debris. Because self-driving cars can travel at high speeds, the development of robust SOD models that can process high-resolution sensor data in real time for long-range situational awareness is a time-compressed task for object detection researchers [1].

Maritime Surveillance

SOD is one of the most important applications in maritime spaces for detecting small vessels, buoys, debris, and individuals overboard. These objects may only exist as a few pixels in reference to a practically infinite dynamic background of water.

- **Drone and ship-based detection**: Both UAVs and ship-based camera systems are used for surveillance. Detecting other drones or small boats from these platforms is a key application for security and situational awareness [47].
- **Challenges**: The maritime environment creates specific challenges, with issues such as wave clutter, sun glare, and weather conditions that easily obscure or mimic small objects. Models need to be resilient to the specific environmental challenges.

8.3. Industrial and Manufacturing Defects

In the modern era of manufacturing, the focus is on the elimination of defects using automated quality control processes, and SOD algorithms are a primary part of the vision-based inspection systems for small defects on product surfaces.

- **Defect targets**: Defects include scratches, cracks, pinholes, and contamination on the surface of any material (e.g., semiconductors, textiles, metals, etc.). These small and sometimes subtle defects are precisely why humans have a difficult time consistently detecting them over long durations in manufacturing environments.
- **Operational benefits**: The operational benefits of SOD systems that utilize computer vision are the objective, repeatable, and high-throughput inspection capabilities that are achievable, and this is especially relevant in wind turbine manufacturing, where automated quality control deep learning-based systems are now used to identify small defects/imperfections in the blades prior to deployment [65]. We all recognize that the cost of failure and maintenance down the road is far greater than the initial cost of identifying potential issues early on, allowing us to be proactive. However, these detection models must achieve high accuracy/precision, as false positives can result in the needless disposal of good products in the inspection processes, while false negatives compromise the quality.

8.4. Medical Imaging and Agriculture

The basic purpose of applying SOD in both medical imaging and agriculture is the recognition of small but significant features in complex biological environments.

Medical Imaging

Finding small features in medical scans is typically the first step in diagnosing diseases. Small features and early-phase pathologies exist in SOD's classic little value reference.

- **Lesion and nodule detection**: SOD models have been trained to identify small lesions via computerized tomography (CT), small polyps in colonoscopy videos, and microcalcifications in mammograms. These small characteristics are often the first indicators of cancer and can be easily missed by the naked eye in early-stage screening assessments.
- **Cellular analysis**: In the area of digital pathology, SOD techniques are used to count or classify small bodies/cells in high-resolution images of tissues, which is valuable for the determination of the disease grade, as well as for research purposes.
- **Architectural developments**: The task of detecting small, often ambiguous lesions has fueled the need for new architectures. Transformer-based models, specifically, have shown potential for the detection of subtle rib fractures, which further support the increase in long-range contextual attention to detection tasks to achieve higher detection performances [75]. Much like agriculture, the field of medical imaging clearly demonstrates the importance of high recall, as not detecting a small malignant lesion in an important aspect negatively affects the patient [67]. Other advancements in object detection leveraging general deep learning will ultimately feed into MOD (medical object detection) improvements [75].

Agriculture (Ground Level)

When one thinks of agricultural SOD applications, aerial drone and satellite imagery typically take center stage. However, there are on-the-ground SOD applications that represent practical use.

- **Pest and disease identification**: Similar to aerial applications, ground robots and/or stationary cameras integrated with SOD models can be utilized to identify small pests, tiny insects, or disease spots on the leaves of plants that are potentially difficult to recognize unless physically observed [63].
- **Automated harvesting**: For some crops with commodification potential, robotic harvesters may leverage vision systems with SOD to identify and localize individual fruits or vegetables for picking, wherein the small-target detection must be accurate. While the future of automation in harvesting may be driven globally, it is vital to realize that small, lightweight sensing networks are developed for integration with mobile harvesting equipment, which does not possess higher computational abilities than those of a not-so-powerful robotic earthworm [25].

8.5. Case Study Performance Summary

Table 2, designed for traceability and comparability, consolidates the application-driven cases discussed in Section 8 using the same quantitative backbone and conventions as our main benchmarking (Section 6): for each domain (remote sensing/UAV, autonomous driving, embedded UAVs under edge constraints, high-resolution industrial inspection, real-time surveillance, and general SOD), we report the AP_S (small-object accuracy) and mAP together with the FPS and deterministic latency (ms = 1000/FPS), plus the hardware/runtime context. The included models—SOD-YOLOv8, Improved-DETR, AUHF-DETR, ESOD-RetinaNet, Pika-YOLOv8n, and MGA-Net—represent the spectrum of accuracy–efficiency tradeoffs that practitioners encounter in real deployments.

Table 2. Consolidated case study results across domains.

Domain	Model	Input (pixels)	AP_S	mAP	FPS	Latency (ms)	Notes	Citation
Remote Sensing/UAV Aerial Imagery	SOD-YOLOv8	640 × 640	33.2	53.8	118	8.5	Spatial–frequency modeling; high AP_S; high accuracy	[38]
Autonomous Driving	Improved-DETR	800 × 800	29.5	48.9	32	31.2	Efficient DETR for driving; accuracy–efficiency tradeoff	[44]
UAV-Embedded (Edge-Constrained)	AUHF-DETR	640 × 1280	28.9	49.5	68	14.7	Lightweight transformer for UAVs; high FPS; strict latency	[39]
Industrial Inspection (High-Resolution)	ESOD-RetinaNet	1200 × 800	25.6	43.1	15	66.7	Efficient high-res processing; favors detail preservation	[55]
Surveillance/Real Time (lightweight YOLO family)	Pika-YOLOv8n	640 × 640	18.5	36.2	30	33.3	Lightweight fusion block; prioritizes FPS over AP_S	[52]
General SOD (multiscale attention)	MGA-Net	640 × 640	30.1	51.2	95	10.5	Multiscale Gaussian attention; strong AP_S with high FPS	[42]

Discussion and Takeaways:

Resolution–AP_S coupling: The highest AP_S values in Table 2 appear in domains that preserve adequate input resolution and exploit multiscale/attention fusion (SOD-YOLOv8, MGA-Net), empirically supporting the central claim that maintaining fine spatial detail upstream of the detection head (and optionally using a P2 head) consistently benefits tiny-object accuracy.

Edge-constrained responsiveness: AUHF-DETR achieves high FPS with low parameter/FLOP budgets, yielding latency suitable for on-board UAV/embedded scenarios. This illustrates the practical tradeoff between real-time responsiveness under strict power/thermal envelopes and the last few points of the AP_S.

Domain-aligned designs: Pika-YOLOv8n favors throughput for surveillance-style streams (real-time emphasis), whereas ESOD-RetinaNet leverages high-resolution pipelines in industrial inspection, where preserving the minute texture is mission-critical despite the heavier computation and latency. Improved-DETR positions itself between accuracy and efficiency for driving scenes where long-range small targets matter, but the platform compute is less constrained than that on edge UAVs.

Reporting practice: Presenting the AP_S, mAP, FPS, latency, and hardware/runtime together—precisely as in Table 2—enables a reproducible, scenario-aware comparison. We recommend that future reports disclose these four metrics jointly and specify deployment conditions to avoid misleading cross-paper contrasts.

9. Environmental Adaptation and Multisensor Fusion

The effectiveness of small-object detection (SOD) models is fundamentally challenged by the dynamic operating environments of the real world. Just as during training, challenges can arise with weather, illumination, and the terrain on which the detection regime is assigned. All of these factors can reduce the detection performance by adding noise, decreasing contrast, or modifying the appearance of an object. To overcome these challenges, recent investigations have examined a two-prong approach: (1) enhancing a model's robustness to changes in environment, and (2) the use of multisensor fusion to implement a more complete, resilient perception system. This section describes modern approaches to environmental adaptation and multisensor fusion, both critically important for the reliable development of SOD systems in applications such as autonomous driving, maritime surveillance, and remote sensing.

9.1. Weather, Terrain, and Illumination Challenges

Adverse weather presents a multifaceted problem for vision-based SOD. The sensors that typically underpin detection systems are standard RGB cameras, which derive its performance based on the ability to discriminate objects while still functioning properly in abnormal situations such as adverse weather.

Adverse weather: Adverse weather—fog, rain, snow, and dust—scatter light, obscure object boundaries, and reduce the visibility of objects, making it virtually impossible to differentiate small objects from the background. Rain can scatter light, dropping pixels off the camera in lens distortion, and snow significantly obscures or alters the color profile, making the identification of pixel values as features or targets unlikely. Research associated with robust 3D object detection states that foggy conditions and snow are among the most significant challenges in the SOD framework. An overly degraded performance produces unreliable single sensor systems [75]. The core of the problem is that the object has high frequencies of information that is lost, obfuscated, or masked in features that are too large to provide nuance or highlight the small.

Variable illumination: Lighting conditions such as low light, glare, or rapid illumination changes, such as when entering or exiting a tunnel, all affect the quality of an image (any object-based detection). Low-light scenarios increase sensor noise and decrease the available contrast, thereby depressing the signal-to-noise ratio. With no draw to a target in the noise space of pixels, there is no detection. Only pixels that have received too much information (e.g., the underside of a semi-truck) will saturate and erase detail for an object that could typically be classified. As such, adaptive algorithms that can alter their input settings as illumination changes are critical to a continual and consistent performance.

Complex terrain and backgrounds: In addition to the environmental conditions, the operating environment can have complex effects. In the context of remote sensing, complex environments with variable textures and colors (e.g., sparse vegetation, rocky outcrops)

can obscure the presence of small targets [76]. Similarly, the presence of background clutter in urban settings provides a high degree of ambiguity (e.g., automated detection of pedestrians in crowded urban streets, or for traffic signs), where the small-object detection (SOD) task may be easily confused with other irrelevant factors of the scene. Successfully distinguishing objects from background clutter is critically important, and this is the performance differentiator for the practical application of SOD.

Algorithmic solutions are being addressed, for example, domain adaptation to train models on the basis of simulated adverse conditions and the development of advanced preprocessing image enhancement steps, or region proposals with architectures that are more robust to a priori input-type perturbations (e.g., RGB to LiDAR). However, trained and sophisticated algorithms (as one might infer) cannot fully rely on a single modal sensor—there are inherent limitations related to the reliance on a single sensor whether one trains the model in the loop or uses a priori knowledge. A growing consensus is that the exploration and consideration of multisensor fusion to achieve the goal of true all-weather, all-scenario perception will likely be the primary practical method for the development of perception systems [77].

9.2. Multisensor Fusion

Multisensor fusion merges data from two or more dissimilar sensors into a more resilient, accurate, complete representation than that achieved with any individual sensor. For SOD, multisensor fusion can detect a target and fuse the weaknesses of individual sensors by taking advantage of the complementary nature of the respective sensors. The most common sensors for the augmentation of RGB cameras are LiDAR, radar, thermal/infrared, and depth sensors.

Fusion strategies: There are different levels at which data can be fused into a final representation:

1. **Early (data-level) fusion**: Raw or minimally processed data from different sensors are combined as input, retaining all of the information; however, it assumes that the sensor calibration and synchronization are perfect and produces a highly dimensional dataset that can be expensive to process.
2. **Intermediate (feature-level) fusion**: Features are extracted independently from each sensor stream and concatenated or fused together in the intermediate layer of a neural network. This is a popular method because it retains information and is inexpensive to process, and it allows the network to learn complex cross-modal correlations.
3. **Late (decision-level) fusion**: Each sensor stream is processed by an independent detection model, and the resulting outputs (e.g., bounding boxes, class probabilities) are fused together at the end, providing modularity and relatively easy implementation; however, it can be less effective because it loses valuable low-level correlations between sensor modalities.

Sensor Modalities and Synergies

- **RGB + LiDAR**: This is one of the most common combinations for autonomous driving applications. LiDAR can provide accurate 3D spatial information (distance, geometry) that is invariant to illumination changes and is comparatively less affected by some weather than RGB cameras. The fusion of LiDAR point clouds and RGB accurately localizes small objects in 3D space, including disambiguating small objects from background clutter [75].
- **RGB + thermal/infrared**: Thermal cameras sense emitted heat radiation instead of reflected light and are particularly useful because they remain effective in low light and at night and can detect animate objects (e.g., pedestrians, animals) against cooler backgrounds, overcoming limitations for 24/7 operation in low-light and obfuscating

conditions (e.g., smoke, light fog). Multiperspective RGB and infrared datasets are being established to facilitate research, with a focus on UAV detection in low-visibility conditions [78,79].

- **RGB + radar**: Radar is exceptionally robust to adverse weather conditions, including heavy rain, fog, and snow. Radar provides precise velocity information and can detect objects at long distances. Although radar data are generally spatially sparse, the fusion of RGB imagery provides valuable object detection and tracking capability in scenarios where vision and LiDAR may fail, which is critical for applications such as adaptive cruise control and early collision warning systems [80].
- **RGB + depth (RGB-D)**: Depth sensors (e.g., stereo cameras, structured light) provide per-pixel depth information that can help the system segment objects from the background and improve scale estimation. This modality fusion is particularly effective in indoor robotics and outdoor applications requiring short-range depth estimation [79].

Adaptive fusion mechanisms: Alongside these hardworking methodologies are relatively new *adaptive* fusion mechanisms, which modify the contribution of each sensor in real time based on the setting rather than a static allocation. For example, SamFusion, a model proposed for 3D detection in adverse weather, learns to rely more heavily on LiDAR data when the visual quality of RGB images is degraded by fog or snow [75]. Adaptive weighting methods in RGB-D fusion have similarly adjusted the contribution of each modality based on lighting, distance, and sensor noise [79]. This intelligent fusion will lead us to resilient perception systems that effectively utilize sensor fusion throughout the world. The fusion of multisensor data likely adds nonlinearity to the feature space that deep learning approaches excel in [81]. For these reasons, we expect the research in this field to continue and to involve neural networks that are designed for adaptive multisensor fusion to enable various scientific fields, with applications ranging from autonomous driving to precision agriculture [77,82].

10. Field Taxonomy and Conceptual Maps

To reflect the information-dense and complex nature of small-object detection across the 2023–2025 time period, we cataloged and represent the knowledge taxonomically and visually. We present a conceptual map of the key research themes and methods and their relationships, followed by a cohesive workflow that guides practitioners from problem definition to deployment.

10.1. Concept Map of SOD in 2023–2025

The state of SOD can be reflected on as a system (or a system of systems), where particular dimensions of the SOD challenge are addressed by different components. The concept map in Figure 6 represents the major foundational pillars of our current standing in modern SOD research and practice.

Overview of the concept map (shown in Figure 6):

- **Core node (central)**: "Small Object Detection (2023–2025)" is the central theme of the concept map.
- **Foundations**: The "Core Challenges" represent the foundational problems that are the basis of all research in the area, such as low resolution, background clutter, scale differences, and data imbalance.
- **Methods and architectures**: The major branch on the concept map presents the main algorithmic solutions. For example, a "Multi-Scale Feature Learning" (with FPN, PANet, etc.) strategy directly addresses feature loss and scale differences. The "Transformer & Attention" strategies in models have made progress toward improving feature representation. The "Context Aware Strategies" have made progress toward addressing

background clutter. Finally, some "YOLO Series Enhancements", such as the addition of a P2 small-object head, are specific architectural responses to low resolution.

- **Data and datasets**: This central pillar is more about the data-centric aspect of SOD research. "Data Augmentation" (both geometric and generative) is a direct response to the data imbalance issue. The field has made progress by developing "Specialized Datasets", which indicate that the quality of data will need to improve before progress can be made.
- **Evaluation and benchmarking**: This node refers to the necessity of evaluation and connects the methods and the data, instead of the simplified evaluation metrics that can be used (both accuracy, such as the AP_S, and efficiency, such as FPS), and the evaluation protocols that allow for fair comparisons.
- **Applications and deployment**: These last two pillars bring us to real-world applications. Applications such as remote sensing and autonomous driving help to categorize the types of use case scenarios and their resulting requirements, and the deployment aspect addresses practical constraints. For example, projects need to be "Edge Computing"-aware, employ model compression, and use multisensor fusion to enhance robustness. This is illustrated on the map via the autonomous driving example and the explicit relationship to multisensor fusion, as this is the application requirement.

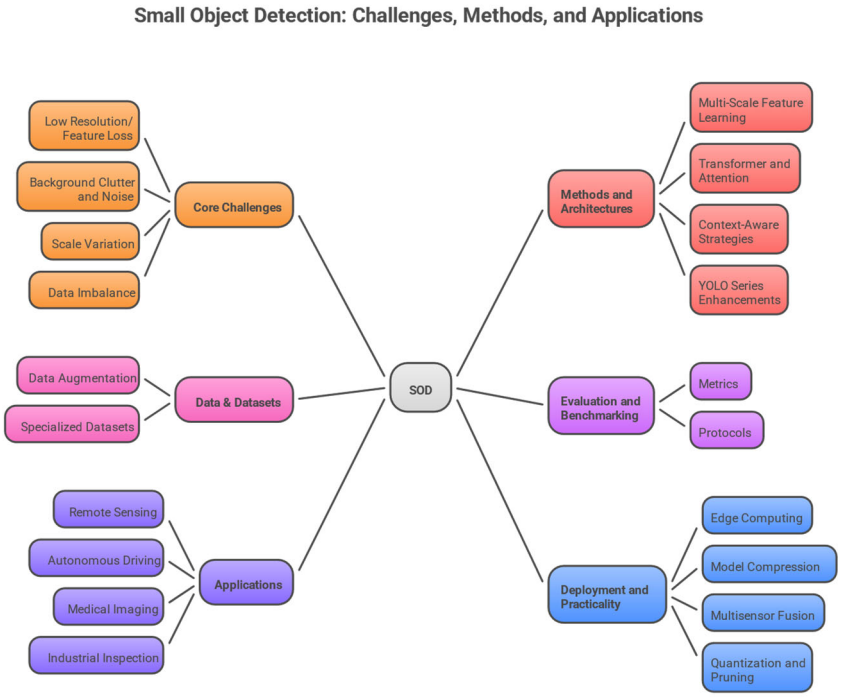

Figure 6. Diagram of Concept map of SOD for 2023–2025.

Overall, the map is meant to provide a comprehensive overview that could suggest all of these SOD research components are not linear progression but rather a symbiotic consideration of the methods, data, evaluation, and recognition of how the core challenges and applications all drive R&D.

10.2. Unified Practical Pipeline

Engineers and researchers hoping to use SOD techniques on a new problem would benefit from a structured roadmap. Figure 7 illustrates a unified pipeline outlining a step-by-step roadmap from initiation to deployment and integrates the key themes reviewed in this survey.

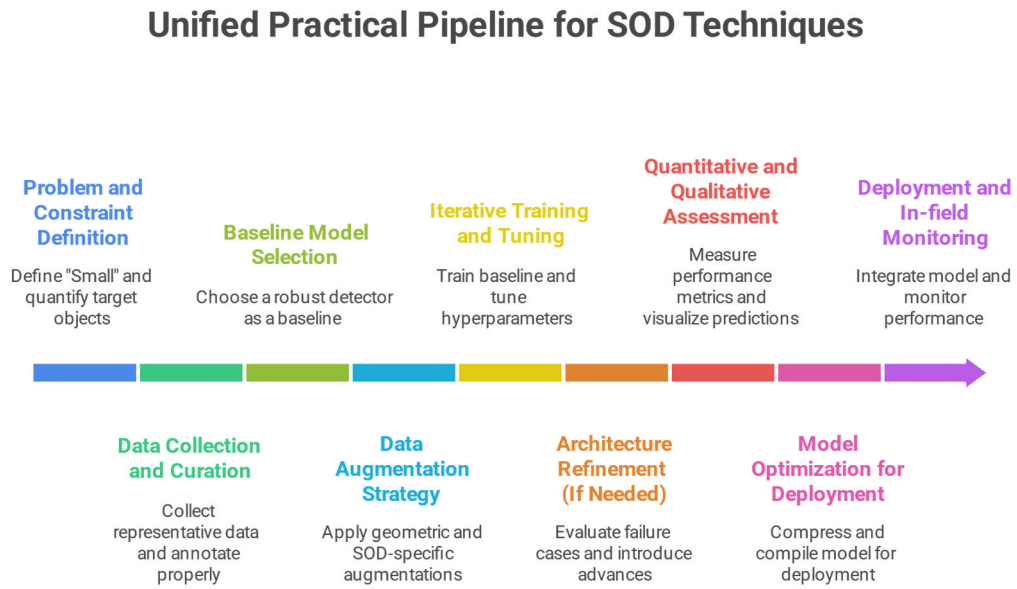

Figure 7. Unified practical pipeline for SOD Techniques.

Pipeline Stages Explained:

1. **Problem and constraint definition:**
   - Define "Small".
   - Quantify the absolute and relative (pixel size) values of the target objects.
   - Identify environment.
   - Characterize the operational conditions (e.g., environment weather, lighting, and background).
   - Define the performance KPIs.
   - Define the initial target metrics (e.g., AP_S > 0.4, Latency < 30 ms, etc.).
   - Specify constraints.
   - Define the hardware (e.g., deployed on a Jetson Orin device), power, memory, etc.
2. **Data collection and curation**:
   - Collect data.
   - Collect representative data from the target environment and use actual sensors if climate/environmental robustness is the goal.
   - Annotate properly.
   - Proper bounding boxes; poor annotation is detrimental to SOD.
   - Create splits.
   - Fixed training, validation, and test splits; these should be maintained throughout group projects, campaigns, etc.
3. **Baseline Model selection**:
   - Start simple.
   - Select a robust and performant detector to start as a baseline (YOLOv8-n/s; YOLOv9-c, etc.).
   - Optional pretrained weights.
   - Pretrained weights on a large-scale dataset, such as COCO, might help as a strong initialization, even if domain-specific.
4. **Data augmentation strategy**:
   - Standard augmentations.
   - Geometric and photometric (scaling, rotation, color jitter, etc.).
   - SOD-specific augmentations.

- Mosaic, MixUp, or copy–paste (to boost small-object frequency/diversity in training scenes).

5. **Iterative training and tuning**:
   - Train the baseline.
   - Train the prior baseline model and establish a performance baseline.
   - Hyperparameter tuning.
   - Tune the key parameters (optimizer, learning rate, loss function weights, etc.), paying close attention to the anchor generation or matching strategy.
6. **Architecture refinement (if needed)**:
   - Evaluate failure cases.
   - Utilize validation set performance from the baseline to identify weaknesses (for example, missed detections in low light, false positives in cluttered scenes, etc.).
   - Introduce advances.
   - Add features based on the above analysis, such as an extra detection head for feature resolution issues (e.g., P2) and attention or context-aware detectors for missing context. Scale variance issues necessitate experimenting with advanced FPN structures (e.g., BiFPN).
7. **Quantitative and qualitative assessment**:
   - Primary metrics.
   - Determine the AP, AP_S, and AR_S performance metrics, defined on the test set.
   - Efficiency metrics.
   - FPS, latency, and resource vs. requirements on target hardware.
   - Qualitative assessment.
   - Visualize predictions against difficult examples; extract analysis on false positives and negatives to inform another iteration and return to consider stage 6 for another iteration if needed; however, move on to stage 8 and optimization if the target metrics are achieved.
8. **Model optimization for deployment**:
   - Compression.
   - Quantization optimizations (e.g., quan8, structured pruning, etc.).
   - Inference engine.
   - Compile the model using an optimized runtime (e.g., TensorRT for NVIDIA GPU).
9. **Deployment and in-field monitoring**:
   - Deploy.
   - Seamlessly integrate the optimized model into the targeted application.
   - Monitor.
   - Continuously monitor the performance in the real word; gather hard cases and additional data to retrain the model at intervals (periodically) or enhance the model overtime in its lifecycle.

As summarized, the final pipeline provides a reasonable approach to addressing SOD, balancing the need for more sophisticated or advanced techniques with a practical, iterative development process aimed at satisfying the basis of any problem—performance metrics and tasks—under real-world constraints.

11. Practical Guidance

Model selection and hands-on playbooks bridging the gap between researcher and practitioner require actionable recommendations. This section presents a model selection decision template based on metrics and case-based playbooks that offers a prescription

for solving SOD challenges and a deployment checklist for ensuring a smooth shift from development to production.

11.1. Model Selection Plan

Selecting the right SOD model is a multidimensional optimization issue. There is no one model that is better for every situation, and the following plan (Figure 8) will help practitioners to identify a starting architecture, in that it assigns common application requirements to model characteristics.

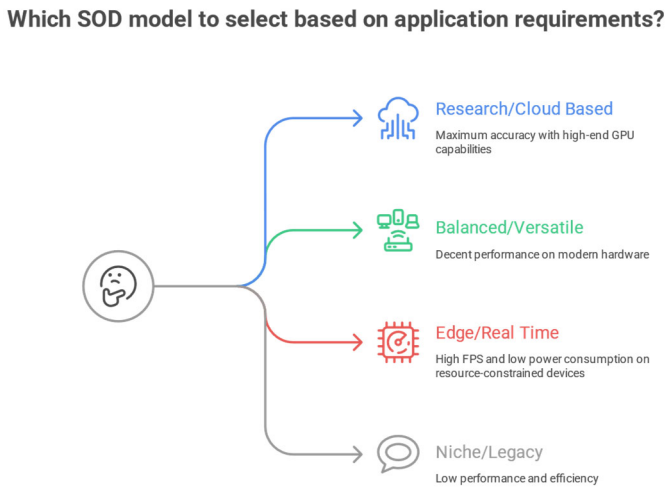

Figure 8. Model selection based on application.

Plan steps:

1. Identify the problem:
   - **Research/cloud based**: Choose this if maximum accuracy is the only concern and use high-end GPU capabilities (A100 or similar). Latency is not a concern. Example: the offline analysis of analyzed satellite imagery.
   - **Balanced/versatile**: This is the sweet spot for many applications that need decent performance on modern hardware (e.g., RTX 40 series) without needing it in strict real time. Example: a manufacturing quality control system.
   - **Edge/real time**: Choose the model for deployment on resource-constrained devices, such as NVIDIA Jetson or FPGAs, where high FPS and low power consumption are key. Example: onboard drone-based detection.
   - **Niche/legacy**: This will generally be avoided, as it represents models with low performances and efficiencies.
2. **Select a starting model**: Select a model that can be used as a baseline for the experiments. If the experiment is edge-based, for example, starting with YOLOv8-S would be a better starting point than starting with a large DETR model.
3. **Consider nuances**: This plan is a guide. Ease of use, support from frameworks, and community documentation are also important. For instance, the YOLO series is very favorable in this regard and, as such, is a common choice for practical implementations.

11.2. Scenario Playbooks

The following playbooks provide prescriptive models that identify steps in engaging common SOD challenges in unique applications.

11.2.1. Playbook 1: Aerial Imagery (UAV/Satellite)

- **Challenge**: Detecting widely dispersed, extremely small-density objects (e.g., cars, people) from high altitudes and multiple perspectives.
- **Playbook**:
- **Data preprocessing**: Utilize tiled inference. Slice high-resolution image (e.g., 8K) into smaller overlapping patches (e.g., 1280 × 1280 pixels due to the zoom-in effect and effectively resizing smaller objects).
- **Model choice**: Choose a model from the "Balanced" or "Research" quadrant (e.g., YOLOv9-C/E). When running inference, you are likely to run offline, and thus, this is more of a concern than accuracy.
- **Architectural tweak**: The model needs to run on high-resolution feature maps for detection. If it is YOLO, make sure it has a P2 head (detecting on a stride-4 feature map), which is non-negotiable for this use case.
- **Augmentation**: Use scale-aware augmentation liberally. Implement copy–paste by adding objects from the dataset to synthetically increase the density of small objects in training scenes. Use Mosaic augmentation with large canvas sizes.
- **Postprocessing**: After you run inference through all tiles, use non-maximum suppression (NMS) or the advanced NMS variant on the joint detections to combine duplicate boxes from overlapping areas.

11.2.2. Playbook 2: Autonomous Driving (Embedded Device)

- **Challenge**: The real-time detection of distance pedestrians, vehicles, and traffic signs on an embedded device (e.g., NVIDIA Jetson Orin) with dynamic weather and lighting conditions.
- **Playbook**:
- **Model choice**: Use a model from the "Edge/Real-Time" quadrant (e.g., YOLOv8-S or RT-DETR-R18). The only constraint is the latency.
- **Sensor fusion**: Do not rely on RGB as a stand-alone sensor; use integrated LiDAR/radar. Use an intermediate fusion approach where camera and LiDAR features are fused in the network backbone to leverage spatial and visual features (thermal camera fusion is also highly recommended for all weather).
- **Optimization**: Use the NVIDIA TensorRT framework for the model conversion and optimization. Use INT8 quantization for fast inference; however, make sure to provide a representative calibration dataset to minimize accuracy loss.
- **Training data**: Ensure the training dataset is composed of diverse scenarios: night, rain, fog, and glare. If real-scene data are limited, leverage simulation environments such as CARLA or generative models to create synthetic examples of these cases to include in the training.
- **Metrics**: In addition to the AP_S, track the end-to-end detection latency (e.g., sensor input to output bounding box) and power consumption of the device.

11.2.3. Playbook 3: Industrial Defect Detection

- **Challenge**: The identification of small, low-contrast defects (scratches, cracks, pinholes, etc.) on uniform surfaces, typically under high-speed production conditions.
- **Playbook**:
- **Environment control**: Before the model is touched, optimize the physical environment. Use controlled, consistent lighting and high-resolution, industrial cameras. Sometimes just changing the angle of the light can make a low-contrast defect highly visible.
- **Data strategy**: This is often a few-shot or one-class problem. You will have many images of normal products and very few defect images.

- **Approach A (detection)**: Use stronger augmentation (copy–paste) to place a visual of the defects onto normal backgrounds to artificially create a larger training set.
- **Approach B (anomaly detection)**: If the defects vary greatly, you can train an anomaly detection model (e.g., PatchCore, PaDiM) on normal samples only. The model will learn to identify anything that deviates from the norm.
- **Model choice**: A lightweight CNN model or a small YOLO model is typically sufficient. The background is simple, so complex context aggregation is often overkill.
- **Input resolution**: Do not downsample original inputs if possible. Run the image through the model at original (native resolution) or tiled to ensure that defect pixels are not lost.

11.3. Deployment Checklist

Before deploying an SOD model into a production solution, work through the following checklist to mitigate common pitfalls:

[ ] 1. Validate the model performance:

- The final model performance (AP_S, AR_S) is evaluated on a held-out test dataset that was not used at any point in the training or tuning.
- The performance is evaluated across important subgroups (e.g., across object classes, day/night, types of weather).

[ ] 2. Benchmark pipeline for efficiency on target hardware:

* [ ] The latency (ms) and throughput (FPS) are measured on the actual hardware deployment (e.g., Jetson AGX Orin; not RTX 4090).

* [ ] The memory usage (VRAM) and power consumption (Watts) are under the device limit.

* [ ] The entire pipeline is benchmarked—including pre- and postprocessing—and not just the forward pass.

[ ] 3. Optimize the model:

* [ ] The model is converted and exported to an optimized format (e.g., TensorRT, ONNX Runtime).

* [ ] Quantization (INT8/FP16) is applied and verified to ensure a tolerable accuracy drop.

* [ ] If further optimization is needed, pruning/distillation should be considered.

[ ] 4. Test robustness and edge cases:

* [ ] The model is tested on adversarial or out-of-distribution data (e.g., corrupted images, unseen environments) to verify that it is not brittle.

* [ ] Failure modes are identified/documented. A further strategy exists to mitigate the failure (e.g., fall-back logic).

[ ] 5. The deployment pipeline is ready:

* [ ] All software dependencies are containerized (e.g., using docker) for portability.

* [ ] Version control is implemented for model, code, and dataset.

* [ ] Monitoring and logging are implemented to track live performance and collect data for future retraining.

12. Discussion: Limitations and Future Directions

In this survey, we systematically reviewed the small-object detection (SOD) advancements from 2023 to 2025, and we have summarized the key contributions to the methods, datasets, and applications. However, despite the many exciting advancements, there are fundamental limitations that persist, as well as many exciting possibilities for future research. In this section, we identify these limitations and discuss important future directions for the next phase of SOD research.

12.1. Current Limitations

There have been major advancements to SOD methods; however, they are not without restrictions. An analysis of some limitations reveals some systematic deficiencies:

1. **Benchmarks and metrics**: Although useful for standardization, the current benchmarks and metrics often do not fully capture the complexities of real-world SOD. For example, for SOD, the most common metric is the average precision for small objects (AP_S), which is used to evaluate performance without consideration to the absolute size, such as 5×5 and 30×30 pixels. As such, the AP_S under the SOD abbreviation masks variation in the assessment, such as a model that was proficient at detecting larger "small" objects, such as 30×30 pixels, or larger small objects, while suffering from the tiny or small "size" of object such as could be defined as 0×0 pixels. Another layer of complexity can be added to evaluate performance based not only on size or standard metrics but also on general-use models that evaluate the practical cost of false positives or negatives, specifically when applied to critically responsible applications such as autonomous driving or medical diagnosis.
2. **Generalization gap**: Most of the recent SOD models usually provide excellent performances on a single benchmark or domain, for example, aerial, but do not extrapolate generalization, especially in adverse conditions or unseen domains. For example, the overfitting on dataset-specific properties such as consistent lighting, the consistent density of objects in the scene, or consistent camera angles; basically, the generalization or non-generalization performances of these models are again a bottleneck for applying them in real-world applications. The transfer of a static and simple dataset to a dynamic and uncertain application is a critical SOD performance hurdle.
3. **Computational cost versus performance tradeoffs**: High-performance SOD models, especially those using high-resolution input images, complex backbone feature fusion, and/or large transformer backbones for implementation, are computationally heavy. There is a tradeoff cost between the performance and deployment efficiency when considering resource-constrained devices such as UAVs or edge or embedded devices where latency and power consumption are critical constraints [83].
4. **Annotated object scarcity bottlenecks and quality**: The performances of deep learning models are mostly tied to available datasets (the quality and quantity of data), and for SOD applications, the severe data scarcity is the bottleneck. Small objects will always benefit from quality, quantity, diversity, and time and costs for their accurate annotation, as this is inherently difficult and time-consuming, especially considering that SOD is always a qualitative approach. Therefore, few-shot or zero-shot SOD is even tougher, given object scarcity at large scales, which makes it difficult to operate "robustly."

12.2. Promising Directions

Given the abovementioned limitations, the application of an innovative thinking process is required to fill the research gaps. The following directions show promise for SOD:

1. **Contextual and explainable AI (XAI) models**: Models of the future should further develop from a simple feature extraction mechanism to improve the contextual understanding of scenes, object–object relationships, and common-sense-like reasoning to reduce ambiguity across SOD applications. Additionally, integrating GNNs or structured knowledge bases when applied in the detection of cluttered-scene objects is particularly useful for resolving ambiguity. Finally, the development of excellent XAI is needed for SOD to debug models that typically fail in real application and would be huge and build trust in critical applications to explain model decision making.
2. **Cross-domain generalization/adaptation**: Future research into UDA and DG, particularly as it relates to SOD methods, will be necessary to balance high benchmark

performance and real performance observation. Future techniques could consider addressing UDA, particularly with detected objects closely to limit labeled data. The detection of new environments by models with limited labeled data could suggest an increase in high-performance SOD methods for practical applications to support speed in constrained applications in autonomous surveillance and environmental monitoring, which are dynamic environments.

3. **Efficient architectures and hardware codesign**: There is an urgent need for new lightweight SOD architectures for edge devices, including exploring neural architecture search (NAS) methods for efficient SOD backbones, current and future model quantization and pruning methods, and hardware codesign wherein algorithms are developed with hardware accelerators for performance and efficiency. Hybrid models that combine CNN- and transformer-based structures, such as RT-DETR, represent exciting developments in this space [83].
4. **Multimodal and multisensor fusion**: The fusion of data from multiple sensors and modalities (e.g., RGB, thermal/infrared, LiDAR, radar) can provide complementary information to surpass the deficiencies of relying on a single type of sensor for the overall objective. For example, thermal data can be used to detect small objects in dark conditions, while LiDAR data can provide depth information. A robust and efficient fusion strategy will be necessary to build trustable SOD systems to operate all day in all weather in domains such as autonomous driving and maritime observation [1].

12.3. The Role of Generative AI and Foundation Models

The growing presence of generative AI and the development of large foundation models will soon transform the field of computer vision, including SOD [84,85].

1. **Data augmentation and synthesis**: Generative models (e.g., diffusion models and GANs) provide a potential solution to the data scarcity problem by generating realistic and varied synthetic training data from which large quantities of sparse or small objects can be created to augment existing data. The result is a more balanced classification distribution and robustness to environmental condition variability.
2. **Foundation models for vision**: Large foundation models (e.g., Vision **Transformers** (ViT) and Vision-Language Models (VLMs)) pretrained on web-scale datasets have exhibited marvelous abilities to learn rich, generalizable visual representations that can be fine-tuned to SOD tasks [86–88], which may lead to significant performance improvements, especially for few-shot or open-world detection [89]. In addition, they may provide more interactive and controllable detection systems that understand semantics from text prompts. The union of generative and discriminative visual foundation models is anticipated to further bolster these capabilities into models that could generate and then detect and ultimately reason about the visual world in a more holistic sense [84,90]. The impact of these powerful models is already being explored in medical imaging, where they have reportedly identified small-scale anomalies, such as rib fractures, with success [91].

13. Conclusions

In this survey, we provide a comprehensive and systematic review of the SOD field during the 2023–2025 period. Employing the PRISMA methodology, we provide a model approach to the field's main challenges, methodological advances and developments, datasets, and benchmarks to advance the state of the art. Our critical taxonomy of methods, including multiscale learning, transformer-based architectures, context-aware strategies, and data augmentations, as well as possible improvements to popular detectors, illustrates

the innovation happening toward overcoming the inherent challenge of detecting small objects with limited visual cues.

We have highlighted the tradeoff between the detection accuracy (AP_S) and computational performance with a comparative analysis of SOD methods to empirically critique the importance of effective decision making in model selection relevant to the context in which it will be employed. The practical guide as a decision matrix and scenario-based playbooks better connect the gap between academics and practitioners to allow researchers to engage with the review's intended purpose: to assist developers as they consider options for operating within their specific use cases.

Despite considerable advancement, SOD remains an ongoing challenge [4]. Persistent issues such as the benchmark–reality gap, the ubiquitous generalization, and data annotation bottlenecks are still present. However, there is excitement ahead. New developments, such as via generative AI for data synthesis and large-scale foundation model adaptation, will drive a new wave of SOD approaches. Along with the continued push for efficient architectures and multisensor fusion, SOD will have new and better systems that are robust, generalizable, practical, and capable of tackling real-world problems to support their increasing complexity. This survey provides important foundational research and catalyzing material for future studies, guiding the community to work toward solving one of the most important, longstanding, and grand challenges in computer vision.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app152211882/s1, PRISMA 2020 Checklist.

Funding: This research received no external funding.

Data Availability Statement: Quantitative table compiled from cited papers; no new datasets or code.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AR	Average Recall
CNN	Convolutional neural network
YOLO	You Only Look Once
COCO	Common Objects in Context
IOU	Intersection over Union
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
FPN	Feature pyramid network
UAV	Unmanned aerial vehicle
VIT	Vision Transformer
DETR	Detection Transformer
AUHF-DETR	Adaptive UAV Hardware-Focused Detection Transformer
GAN	Generative Adversarial Network
AP	Average precision
mAP	Mean average precision
FPS	Frames Per Second
GFLOP	Giga Floating Point Operations per Second
BiFPN	Bidirectional Feature Pyramid Network
PANet	Path Aggregation Network
RAM	Random Access Memory
SOD	Small object detection
GNN	Graph Neural Network

NAS	Neural architecture search
VLM	Vision Language Model
XAI	Explainable Artificial Intelligence
NMS	Non-maximum suppression
LiDAR	Light Detection and Ranging

References

1. Wang, J.; Su, J. A review of object detection techniques in IoT-based intelligent transportation systems. *Comput. Mater. Contin.* **2025**, *84*, 125–152. [CrossRef]
2. Muzammul, M.; Li, X. Comprehensive review of deep learning-based tiny object detection: Challenges, strategies, and future directions. *Knowl. Inf. Syst.* **2025**, *67*, 3825–3913. [CrossRef]
3. Khan, Z.; Shen, Y.; Liu, H. ObjectDetection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions. *Agriculture* **2025**, *15*, 1351. [CrossRef]
4. Iqra; Giri, K.J.; Javed, M. Small object detection in diverse application landscapes: A survey. *Multimed. Tools Appl.* **2024**, *83*, 88645–88680. [CrossRef]
5. Mu, J.; Su, Q.; Wang, X.; Liang, W.; Xu, S.; Wan, K. A small object detection architecture with concatenated detection heads and multi-head mixed self-attention mechanism. *J. Real-Time Image Process.* **2024**, *21*, 184. [CrossRef]
6. Memari, M.; Shakya, P.; Shekaramiz, M.; Seibi, A.C.; Masoum, M.A.S. Review on the advancements in wind turbine blade inspection: Integrating drone and deep learning technologies for enhanced defect detection. *IEEE Access* **2024**, *12*, 33236–33282. [CrossRef]
7. Liu, X.; Liu, B. EGFE-Net: An edge-guided and feature elimination network for small object detection. *Expert Syst. Appl.* **2025**, *299 Pt A*, 129989. [CrossRef]
8. Nikouei, M.; Baroutian, B.; Nabavi, S.; Taraghi, F.; Aghaei, A.; Sajedi, A.; Ebrahimi Moghaddam, M. Small object detection: A comprehensive survey on challenges, techniques and real-world applications. *Intell. Syst. Appl.* **2025**, *27*, 200561. [CrossRef]
9. Tian, J.; Jin, Q.; Wang, Y.; Yang, J.; Zhang, S.; Sun, D. Performance analysis of deep learning-based object detection algorithms on COCO benchmark: A comparative study. *J. Eng. Appl. Sci.* **2024**, *71*, 76. [CrossRef]
10. Yang, J.; Zhang, X.; Song, C. Research on a small target object detection method for aerial photography based on improved YOLOv7. *Vis. Comput.* **2025**, *41*, 3487–3501. [CrossRef]
11. Chen, Z.; Ji, H.; Zhang, Y.; Zhu, Z.; Li, Y. High-resolution feature pyramid network for small object detection on drone view. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 475–489. [CrossRef]
12. Zhao, Z.; Du, J.; Li, C.; Fang, X.; Xiao, Y.; Tang, J. Dense tiny object detection: A scene context guided approach and a unified benchmark. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5606913. [CrossRef]
13. Zhu, H.; Xu, C.; Yang, W.; Zhang, R.; Zhang, Y.; Xia, G.S. Robust tiny object detection in aerial images amidst label noise. *arXiv* **2024**, arXiv:2401.08056. [CrossRef]
14. Rao, M.K.; Kumar23, P.A. Exploring the advancements and challenges of object detection in video surveillance through deep learning: A systematic literature review and outlook. *J. Theor. Appl. Inf. Technol.* **2025**, *103*, 6.
15. Dalal, M.; Mittal, P. A systematic review of deep learning-based object detection in agriculture: Methods, challenges, and future directions. *Comput. Mater. Contin.* **2025**, *84*, 57–91. [CrossRef]
16. Albuquerque, C.; Henriques, R.; Castelli, M. Deep learning-based object detection algorithms in medical imaging: Systematic review. *Heliyon* **2025**, *11*, e41137. [CrossRef] [PubMed]
17. Gonsalves, A.P.; Yadav, R.K. A systematic review of pedestrian detection techniques using deep learning. *Intell. Comput. Commun. Tech.* **2025**, *1*, 448–455.
18. Sun, Y.; Zhang, C.; Li, X.; Jing, X.; Kong, H.; Wang, Q.-G. MDSF-YOLO: Advancing object detection with a multiscale dilated sequence fusion network. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *early access*. 1–12. [CrossRef] [PubMed]
19. Mao, Y.; Zhang, H.; Li, R.; Zhu, F.; Sun, R.; Ji, P. HSF-DETR: Hyper Scale Fusion Detection Transformer for Multi-Perspective UAV Object Detection. *Remote Sens.* **2025**, *17*, 1997. [CrossRef]
20. Lai, D.; Kang, K.; Xu, K.; Ma, X.; Zhang, Y.; Huang, F.; Chen, J. Enhancing UAV object detection with an efficient multi-scale feature fusion framework. *PLoS ONE* **2025**, *20*, e0332408. [CrossRef]
21. Tang, Z.; Fang, L.; Sun, S.; Gong, Y.; Li, Q. ML-DETR: Multiscale-Lite Detection Transformer for identification of mature cherry tomatoes. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2547018. [CrossRef]
22. Zhang, H.; Xiao, P.; Yao, F.; Zhang, Q.; Gong, Y. Fusion of multi-scale attention for aerial images small-target detection model based on PARE-YOLO. *Sci. Rep.* **2025**, *15*, 4753. [CrossRef]
23. Wu, Z.; Zhen, H.; Zhang, X.; Bai, X.; Li, X. SEMA-YOLO: Lightweight Small Object Detection in Remote Sensing Image via Shallow-Layer Enhancement and Multi-Scale Adaptation. *Remote Sens.* **2025**, *17*, 1917. [CrossRef]

24. Liu, Y.; Zhao, J.; Xu, C.; Hou, Y.; Jiang, Y. YOLO-Pika: A lightweight improved model of YOLOv8n incorporating Fusion_Block and multi-scale fusion FPN and its application in the precise detection of plateau pikas. *Front. Plant Sci.* **2025**, *16*, 1607492. [CrossRef]
25. Sundaralingam, H. Advancing Object Detection Models: An Investigation Focused on Small Object Detection in Complex Scenes. Ph.D. Thesis, Lakehead University, Thunder Bay, ON, Canada, 2025.
26. Tong, Y.; Ye, H.; Yang, J.; Yang, X. ACD-DETR: Adaptive Cross-Scale Detection Transformer for Small Object Detection in UAV Imagery. *Sensors* **2025**, *25*, 5556. [CrossRef] [PubMed]
27. Yu, Y.; Huang, M.; Wang, K.; Tang, X.; Bao, J.; Fan, Y. LS-YOLO: A lightweight small-object detection framework with region scaling loss and self-attention for intelligent transportation systems. *Signal Image Video Process.* **2025**, *19*, 1005. [CrossRef]
28. Rekavandi, A.M.; Rashidi, S.; Boussaid, F.; Hoefs, S.; Akbas, E.; Bennamoun, M. Transformers in small object detection: A benchmark and survey of state-of-the-art. *ACM Comput. Surv.* **2025**, *58*, 64. [CrossRef]
29. Ding, G.; Liu, J.; Li, D.; Fu, X.; Zhou, Y.; Zhang, M.; Li, W.; Wang, Y.; Li, C.; Geng, X. A Cross-Stage Focused Small Object Detection Network for Unmanned Aerial Vehicle Assisted Maritime Applications. *J. Mar. Sci. Eng.* **2025**, *13*, 82. [CrossRef]
30. Patel, R.; Chandalia, D.; Nayak, A.; Jeyabose, A.; Jijo, D. CGI-based synthetic data generation and detection pipeline for small objects in aerial imagery. *IEEE Access* **2025**, *13*, 61192–61206. [CrossRef]
31. Chen, Y.; Yan, Z.; Zhu, Y. A unified framework for generative data augmentation: A comprehensive survey. *arXiv* **2023**, arXiv:2310.00277. [CrossRef]
32. Nisa, U.; Pozi, M.S.M.; Saip, M.A. A decade of research in small object detection: A comprehensive bibliometric analysis. *Int. J. Data Sci. Anal.* **2025**, *20*, 7331–7355. [CrossRef]
33. Li, Y.; Dong, X.; Chen, C.; Zhuang, W.; Lyu, L. A simple background augmentation method for object detection with diffusion model. In *Lecture Notes in Computer Science*; Computer Vision—ECCV 2024; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Springer: Cham, Switzerland, 2025; p. 15124. [CrossRef]
34. Nisa, U. Image augmentation approaches for small and tiny object detection in aerial images: A review. *Multimed. Tools Appl.* **2025**, *84*, 21521–21568. [CrossRef]
35. Liu, X.; Luo, X.; Ye, Y.; Huang, X. Potential of diffusion-generated data on salient object detection. *IEEE Trans. Multimed.* **2025**, 1–13. [CrossRef]
36. Li, P.; Zhang, T.; Qing, C.; Zhang, S. F^2SOD: A Federated Few-Shot Object Detection. *Electronics* **2025**, *14*, 1651. [CrossRef]
37. Ferreira, A.D.S.; Ramos, A.P.M.; Junior, J.M.; Gonçalves, W.N. Data augmentation and resolution enhancement using GANs and diffusion models for tree segmentation. *arXiv* **2025**, arXiv:2505.15077. [CrossRef]
38. Li, Y.; Li, Q.; Pan, J.; Zhou, Y.; Zhu, H.; Wei, H.; Liu, C. SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images. *Remote Sens.* **2024**, *16*, 3057. [CrossRef]
39. Zhu, M.; Zhong, H.; Zhao, C.; Du, Z.; Huang, Z.; Liu, M.; Chen, H.; Zou, C.; Chen, J.; Yang, M.; et al. Active-O3: Empowering multimodal large language models with active perception via GRPO. *arXiv* **2025**, arXiv:2505.21457. [CrossRef]
40. Feng, C.; Zhong, Y.; Jie, Z.; Xie, W.; Ma, L. InstaGen: Enhancing object detection by training on synthetic dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, DC, USA, 17–21 June 2024; pp. 14121–14130.
41. Flores-Calero, M.; Astudillo, C.A.; Guevara, D.; Maza, J.; Lita, B.S.; Defaz, B.; Ante, J.S.; Zabala-Blanco, D.; Armingol Moreno, J.M. Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. *Mathematics* **2024**, *12*, 297. [CrossRef]
42. Jing, S.; Guo, G.; Xu, X.; Zhao, Y.; Wang, H.; Lv, H.; Feng, Y.; Zhang, Y. ESVT: Event-based streaming vision transformer for challenging object detection. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5607113. [CrossRef]
43. Tang, D.; Tang, S.; Wang, Y.; Guan, S.; Jin, Y. A global object-oriented dynamic network for low-altitude remote sensing object detection. *Sci. Rep.* **2025**, *15*, 19071. [CrossRef]
44. Liu, D.; Zhang, J.; Qi, Y.; Xi, Y.; Jin, J. Exploring lightweight structures for tiny object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5623215. [CrossRef]
45. Shi, S.; Fang, Q.; Xu, X.; Dong, D. Multiscale Gaussian attention mechanism for tiny-object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5635216. [CrossRef]
46. Liu, Z.; Jiang, H.; Zhong, T.; Wu, Z.; Ma, C.; Li, Y.; Yu, X.; Zhang, Y.; Pan, Y.; Shu, P.; et al. Holistic evaluation of GPT-4V for biomedical imaging. *arXiv* **2023**, arXiv:2312.05256. [CrossRef]
47. Rekavandi, A.M.; Xu, L.; Boussaid, F.; Seghouane, A.-K.; Hoefs, S.; Bennamoun, M. A guide to image- and video-based small object detection using deep learning: Case study of maritime surveillance. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 2851–2879. [CrossRef]
48. Farooq, M.A.; Shariff, W.; O'Callaghan, D.; Merla, A.; Corcoran, P. On the role of thermal imaging in automotive applications: A critical review. *IEEE Access* **2023**, *11*, 25152–25173. [CrossRef]

49. Costa, D.; Silva, C.; Costa, J.; Ribeiro, B. Enhancing pest detection models through improved annotations. In *Progress in Artificial Intelligence, Proceedings of the 22nd EPIA Conference on Artificial Intelligence, EPIA 2023, Faial Island, Azores, 5–8 September 2023*; Moniz, N., Vale, Z., Cascalho, J., Silva, C., Sebastião, R., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; p. 14116. [CrossRef]
50. Nie, Y.; Fang, C.; Cheng, L.; Lin, L.; Li, G. Adapting object size variance and class imbalance for semi-supervised object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington DC, USA, 7–14 February 2023; pp. 1966–1974. [CrossRef]
51. Mudavath, T.; Mamidi, A. Object detection challenges: Navigating through varied weather conditions—A comprehensive survey. *J. Ambient Intell. Humaniz. Comput.* **2025**, *16*, 443–457. [CrossRef]
52. Jiang, Y.; Yan, X.; Ji, G.-P.; Fu, K.; Sun, M.; Xiong, H.; Fan, D.-P.; Khan, F.S. Effectiveness assessment of recent large vision–language models. *Vis. Intell.* **2024**, *2*, 17. [CrossRef]
53. Nie, J.; Wang, Y.; Yu, Z.; Zhou, S.; Lei, J. High-precision grain size analysis of laser-sintered Al_2O_3 ceramics using a deep-learning-based ceramic grains detection neural network. *Comput. Mater. Sci.* **2025**, *250*, 113724. [CrossRef]
54. Guo, H.; Wu, Q.; Wang, Y. AUHF-DETR: A Lightweight Transformer with Spatial Attention and Wavelet Convolution for Embedded UAV Small Object Detection. *Remote Sens.* **2025**, *17*, 1920. [CrossRef]
55. Han, J.; Sun, F.; Hou, Y.; Sun, J.; Li, H. Exploring a lightweight and efficient network for salient object detection in ORSI. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5631014. [CrossRef]
56. Liu, J.; Wang, Y.; Cao, Y.; Guo, C.; Shi, P.; Li, P. Unified Spatial-Frequency Modeling and Alignment for Multi-Scale Small Object Detection. *Symmetry* **2025**, *17*, 242. [CrossRef]
57. Zhao, H.; Zhang, S.; Peng, X.; Lu, Z.; Li, G. Improved object detection method for autonomous driving based on DETR. *Front. Neurorobotics* **2025**, *18*, 1484276. [CrossRef] [PubMed]
58. Liu, K.; Fu, Z.; Jin, S.; Chen, Z.; Zhou, F.; Jiang, R.; Chen, Y.; Ye, J. ESOD: Efficient small object detection on high-resolution images. *IEEE Trans. Image Process.* **2025**, *34*, 183–195. [CrossRef] [PubMed]
59. Qian, B.; Qian, J.; Wen, Z.; Wu, D.; He, S.; Chen, J.; Ranjan, R. DEEPCON: Improving distributed deep learning model consistency in edge–cloud environments via distillation. *IEEE Trans. Cogn. Commun. Netw.* **2025**. [CrossRef]
60. Radulov, N.; Zhang, Y.; Bujanca, M.; Ye, R.; Luján, M. A framework for reproducible benchmarking and performance diagnosis of SLAM systems. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 14225–14232. [CrossRef]
61. Khan, A.; Ullah, H.; Munir, A. LiteCOD: Lightweight Camouflaged Object Detection via Holistic Understanding of Local-Global Features and Multi-Scale Fusion. *AI* **2025**, *6*, 197. [CrossRef]
62. Sheikholeslami, S.; Ghasemirahni, H.; Payberah, A.H.; Wang, T.; Dowling, J.; Vlassov, V. Utilizing large language models for ablation studies in machine learning and deep learning. In Proceedings of the 5th Workshop on Machine Learning and Systems (EuroMLSys 2025), World Trade Center, Rotterdam, The Netherlands, 31 March 2025; Association for Computing Machinery: New York, NY, USA, 2025; pp. 230–237. [CrossRef]
63. Sharma, L.; Rihan, M.; Rana, N.K.; Dube, S.K.; Asgher, M.S. Advanced modeling of forest fire susceptibility and sensitivity analysis using hyperparameter-tuned deep learning techniques in the Rajouri district, Jammu and Kashmir. *Adv. Space Res.* **2025**, *76*, 614–632. [CrossRef]
64. Bong, H.M.; de Azambuja, R.; Beltrame, G. BlabberSeg: Real-time embedded open-vocabulary aerial segmentation. *arXiv* **2024**, arXiv:2410.12979. [CrossRef]
65. Yang, Z.; Khan, Z.; Shen, Y.; Liu, H. GTDR-YOLOv12: Optimizing YOLO for Efficient and Accurate Weed Detection in Agriculture. *Agronomy* **2025**, *15*, 1824. [CrossRef]
66. Gao, X.; Gao, J.; Qureshi, W.A. Applications, Trends, and Challenges of Precision Weed Control Technologies Based on Deep Learning and Machine Vision. *Agronomy* **2025**, *15*, 1954. [CrossRef]
67. Lokhande, H.; Ganorkar, S.R. Object detection in video surveillance using MobileNetV2 on resource-constrained low-power edge devices. *Bull. Electr. Eng. Inform.* **2025**, *14*, 357–365. [CrossRef]
68. Xiao, L.; Li, W.; Yao, S.; Liu, H.; Ren, D. High-precision and lightweight small-target detection algorithm for low-cost edge intelligence. *Sci. Rep.* **2024**, *14*, 23542. [CrossRef]
69. Ayoub, K. Dissertation Submitted to the Department of Computer Science in Partial Fulfillment of the Requirements for Engineer's Degree in Computer Science Specialty. Engineer's Thesis, Higher School of Computer Science, Amizour, Algeria, 2025. [CrossRef]
70. Surantha, N.; Sutisna, N. Key Considerations for Real-Time Object Recognition on Edge Computing Devices. *Appl. Sci.* **2025**, *15*, 7533. [CrossRef]
71. Raza, S.M.; Abidi, S.M.H.; Masuduzzaman, M.; Shin, S.Y. Survey on the application with lightweight deep learning models for edge devices. *TechRxiv* **2025**, preprint. [CrossRef]
72. Li, J.; Zhang, T.; Luo, Q.; Zeng, S.; Luo, X.; Chen, C.L.P.; Yang, C. A lightweight palm fruit detection network for harvesting equipment integrates binocular depth matching. *Comput. Electron. Agric.* **2025**, *233*, 110061. [CrossRef]

73. Gu, Y.; Jing, Y.; Li, H.-D.; Shi, J.; Lin, H. DEMNet: A Small Object Detection Method for Tea Leaf Blight in Slightly Blurry UAV Remote Sensing Images. *Remote Sens.* **2025**, *17*, 1967. [CrossRef]
74. Zhang, Z.; Shu, Z. Unmanned Aerial Vehicle (UAV)-Assisted Damage Detection of Wind Turbine Blades: A Review. *Energies* **2024**, *17*, 3731. [CrossRef]
75. Palladin, E.; Dietze, R.; Narayanan, P.; Bijelic, M.; Heide, F. SAMFusion: Sensor-adaptive multimodal fusion for 3D object detection in adverse weather. In *Computer Vision—ECCV 2024, Proceedings of the 18th European Conference, Milan, Italy, 29 September–4 October 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15119, pp. 484–503. [CrossRef]
76. Platel, A.; Sandino, J.; Shaw, J.; Bollard, B.; Gonzalez, F. Advancing Sparse Vegetation Monitoring in the Arctic and Antarctic: A Review of Satellite and UAV Remote Sensing, Machine Learning, and Sensor Fusion. *Remote Sens.* **2025**, *17*, 1513. [CrossRef]
77. Zhang, X.; Li, J.; Li, Z.; Liu, H.; Zhou, M.; Wang, L.; Zou, Z. *Multi-Sensor Fusion for Autonomous Driving*; Springer: Singapore, 2023. [CrossRef]
78. Tavaris, D.; de Zan, A.; Toma, A.; Foresti, G.L.; Scagnetto, I.; Martinel, N. Multi-perspective RGB and infrared dataset for UAV detection. *IEEE Access* **2025**, *13*, 168792–168803. [CrossRef]
79. Brenner, M.; Reyes, N.H.; Susnjak, T.; Barczak, A.L.C. RGB-D and thermal sensor fusion: A systematic literature review. *IEEE Access* **2023**, *11*, 82410–82442. [CrossRef]
80. Meydani, A. State-of-the-art analysis of the performance of the sensors utilized in autonomous vehicles in extreme conditions. In *Artificial Intelligence and Smart Vehicles ICAISV 2023*; Communications in Computer and Information Science; Ghatee, M., Hashemi, S.M., Eds.; Springer: Cham, Switzerland, 2023; Volume 1883, pp. 137–166. [CrossRef]
81. Sharma, M. Multimodal Data Fusion and Model Compression Methods for Computer Vision. Ph.D. Thesis, Rochester Institute of Technology, Rochester, NY, USA, 2024.
82. Cai, Z.; Wen, C.; Bao, L.; Ma, H.; Yan, Z.; Li, J.; Gao, X.; Yu, L. Fine-Scale Grassland Classification Using UAV-Based Multi-Sensor Image Fusion and Deep Learning. *Remote Sens.* **2025**, *17*, 3190. [CrossRef]
83. El Zeinaty, C.; Hamidouche, W.; Herrou, G.; Menard, D. Designing object detection models for TinyML: Foundations, comparative analysis, challenges, and emerging solutions. *ACM Comput. Surv.* **2025**, *58*, 50. [CrossRef]
84. Liu, X.; Zhou, T.; Wang, C.; Wang, Y.; Wang, Y.; Cao, Q.; Du, W.; Yang, Y.; He, J.; Qiao, Y.; et al. Toward the unification of generative and discriminative visual foundation model: A survey. *Vis. Comput.* **2025**, *41*, 3371–3412. [CrossRef]
85. Giacalone, E. AI-Powered Autonomous Industrial Monitoring: Integrating Robotics, Computer Vision, and Generative AI. Ph.D. Thesis, Politecnico di Torino, Turin, Italy, 2025. Available online: https://webthesis.biblio.polito.it/35371/ (accessed on 15 October 2025).
86. Edozie, E.; Shuaibu, A.N.; John, U.K.; Sadiq, B.O. Comprehensive review of recent developments in visual object detection based on deep learning. *Artif. Intell. Rev.* **2025**, *58*, 277. [CrossRef]
87. Awais, M.; Naseer, M.; Khan, S.; Anwer, R.M.; Cholakkal, H.; Shah, M.; Yang, M.H.; Khan, F.S. Foundation models defining a new era in vision: A survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2245–2264. [CrossRef] [PubMed]
88. Guleria, A.; Varshney, K.; Jindal, S. A systematic review: Object detection. *AI Soc.* **2025**, 1–18. [CrossRef]
89. Sapkota, R.; Roumeliotis, K.I.; Cheppally, R.H.; Flores Calero, M.; Karkee, M. A review of 3D object detection with vision–language models. *arXiv* **2025**, arXiv:2504.18738. [CrossRef]
90. Hussain, A.; Ali, S.; Farwa, U.E.; Mozumder, M.A.I.; Kim, H.-C. Foundation models: From current developments, challenges, and risks to future opportunities. In Proceedings of the 27th International Conference on Advanced Communications Technology (ICACT), Pyeong Chang, Republic of Korea, 16–19 February 2025; pp. 51–58. [CrossRef]
91. Saraei, M.; Lalinia, M.; Lee, E.-J. Deep learning-based medical object detection: A survey. *IEEE Access* **2025**, *13*, 53019–53038. [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.