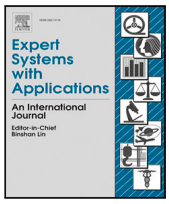
Expert Systems With Applications 280 (2025) 127447

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



SO-YOLOv8: A novel deep learning-based approach for small object detection with YOLO beyond COCO

Iqra, Kaisar Javeed Giri *

Department of Computer Science, Islamic University of Science and Technology (IUST), Awantipora, Pulwama, 192122, J&K, India

ARTICLE INFO

Keywords:
Computer vision
Deep learning
Optimization
Precision
Small object detection
YOLOv8

ABSTRACT

Small object detection in images is a significant challenge in computer vision due to issues like low resolution, occlusion, and scale variation, often resulting in existing models missing important details or requiring complex, large-scale setups. This paper introduces SO-YOLOv8, an enhanced version of the YOLO model that focuses on small object detection. The proposed model uses advanced hyperparameter optimization, sophisticated data augmentation, and multi-scale training to improve detection accuracy. SO-YOLOv8 also includes a Squeeze-and-Excitation (SE) block, which helps the model better recognize features of small objects. Experimental results on the PASCAL VOC 2012 dataset, a benchmark known for diverse and challenging object scales, demonstrate substantial improvements, achieving a **precision of 1.0, showing an increase of 6%** and an enhanced **mean Average Precision (mAP) score of 0.79, reflecting a 1% increase in mAP** compared to YOLOv8. While the mAP gain may seem marginal, even a slight improvement in small object detection significantly impacts real-world applications such as autonomous vehicles (detecting distant pedestrians or small road hazards), surveillance and security (identifying concealed objects in crowded environments), medical imaging (spotting small anomalies like tumors), and remote sensing (detecting small objects in satellite or drone imagery). Also, the 6% increase in precision indicates a significant reduction in false positives, making the detection system more reliable and reducing misclassifications that could otherwise lead to critical errors. These findings confirm that targeted customization of YOLO's architecture can effectively address the challenges associated with small object detection. This research contributes to the ongoing development of object detection methodologies and establishes a robust foundation for future work in small object detection.

1. Introduction

Object detection (Zou, Chen, Shi, Guo, & Ye, 2023) is a key task in the computer vision (Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018) field that involves classifying objects and localizing them within images. Typically, it follows a two-step process: first, identifying the likely locations of target objects, and second, classifying those objects into their corresponding categories. Despite significant advancements in this field, the challenge of detecting small objects (Chen et al., 2020) remains formidable. As methods such as R-CNN, Fast R-CNN, and Faster R-CNN depend on region proposals for detecting objects, they attain high accuracy but experience significant computational requirements and prolonged inference durations (Girshick, 2015; Girshick, Donahue, Darrell, & Malik, 2014; Ren, He, Girshick, & Sun, 2016). These methods are inappropriate for real-time applications, particularly in contexts necessitating high responsiveness, such as autonomous driving and monitoring. The YOLO (You Only Look Once) series (Du, 2018) transformed object detection by conceptualizing it as a singular regression issue, facilitating real-time processing. Still, these models are not as good at finding small objects because of things like the loss of fine-grained features *(the downsampling process in the network's shallow layers often gets rid of small objects, making their features indistinguishable)*, the dominance of larger objects *(when features are extracted, larger objects often dominate smaller ones, making their detection less accurate)*, and scale variation *(YOLO models have trouble with the different size scales of items, making them less precise and memory-efficient for smaller objects)*. These limitations underscore the necessity for architectural improvements, including feature recalibration, multi-scale training, and sophisticated augmentation techniques. The proposed SO-YOLOv8 solves these issues by adding a Squeeze-and-Excitation (SE) block, re-calibrated anchor boxes, and techniques for adding more data to help find small objects. Small object detection poses unique challenges due to their limited size, lower resolution (Tong, Wu, & Zhou, 2020), and less distinguishable features compared to larger objects (Zhang, Zhang, Huang, Han, & Zhao, 2024). This often leads

* Corresponding author.
E-mail addresses: iqra@iust.ac.in (Iqra), kaisar.giri@islamicuniversity.edu.in (K.J. Giri).

https://doi.org/10.1016/j.eswa.2025.127447
Received 5 November 2024; Received in revised form 22 March 2025; Accepted 25 March 2025
Available online 6 April 2025

to overlooked detections and reduced accuracy, necessitating models with specialized feature extraction capabilities to address these limitations effectively. Zhu, Liang, Zhang, Huang, Li et al. (2016) defined small objects as the objects that occupy less than 20% of an image specifically in the context of traffic sign datasets. Traffic sign is considered small if the bounding box's width is less than 20% of the image's width, and its height is less than the image's total height. Similarly, Torralba, Fergus, and Freeman (2008) defined small objects as objects with dimensions equal to or smaller than 32 × 32 pixels. Also, the small object dataset (Chen, Liu, Tuzel, & Xiao, 2017) describes objects as small when the mean relative overlap, the ratio of the bounding box area to the image area ranges from 0.08%–0.58%, corresponding to object sizes between 16 × 16 and 42 × 42 pixels in a VGA image. In this work, we adopt these definitions, as they are well-established and commonly referenced in small object detection research. The definition also aligns with the characteristics of the Pascal VOC 2012 dataset, where small objects typically occupy less than 10% of the total image area. These well-established criteria ensure that the evaluation of SO-YOLOv8 is consistent with recognized benchmarks in object detection. The group of models, called YOLO (You Only Look Once) (Diwan, Anirudh, & Tembhurne, 2023), has become famous for detecting objects because these models exhibit exceptional efficiency by performing object detection tasks in a single iteration through the network. However, although they demonstrate exceptional performance in identifying medium and big-sized objects, the task of detecting small objects presents considerable difficulty, frequently resulting in overlooked detections or less than satisfactory accuracy. In models like YOLOv8 (Farooq, Muaz, Khan Jadoon, Aafaq, & Khan, 2024), important details needed to detect these small objects can be lost in the shallow layers of the network. Also, during feature extraction, small objects may be overshadowed by larger ones, leading to the loss of key information. Solving these problems is important to improve the accuracy and reliability of detection, especially in real-world situations where detecting small objects is crucial. Detection of small objects, such as faces or license plates (Suhartono, Zain, & Ardilla, 2024), is critical in surveillance for security and monitoring purposes. Similarly, in the field of medical imaging, the identification of early-stage small cancers (Rahimi, Mostafavi, & Arabameri, 2024) can be life-saving. Safe navigation in autonomous vehicles (Mahaur, Mishra, & Kumar, 2023; Özcan, Altun, & Parlak, 2024) is contingent upon the proper detection of small objects, such as road signs and pedestrians. Thus, enhancing small object detection methods can have significant ramifications in several domains. The conventional YOLO models, although successful in general object detection, typically encounter difficulties in detecting small objects (Xianbao, Guihua, Yu, & Zhaomin, 2021) because of constraints in extracting features at lower resolutions. As a result, smaller objects have lower precision and recall rates. The task is to develop a model that can consistently achieve true detection accuracy for objects of different sizes, particularly those at the smaller end of the range. YOLO models are primarily trained and assessed on COCO (Common Objects in Context) dataset which is a prominent benchmark in object detection, providing more than 200,000 annotated images spanning 80 object categories. This work demonstrates that SO-YOLOv8 is optimized for small object detection and is evaluated on the Pascal VOC 2012 dataset, showcasing its efficacy beyond the traditional COCO benchmark. This represents progress in adapting YOLO-based models to datasets characterized by varied object scales and distributions. This paper addresses the challenge of small object detection by optimizing the YOLOv8x model through several key upgrades:

- Small object detection is enhanced by focusing on finer details, allowing for more accurate recognition of small targets.
- An SE block (Squeeze-and-Excitation) is incorporated into the YOLOv8x backbone, enhancing feature learning specifically for small object detection.
- Target detection is optimized to increase processing speed and improve overall efficiency.
- Advanced feature fusion techniques are integrated, enabling the model to capture features across multiple scales more effectively.
- Data augmentation techniques are employed to significantly improve the model's robustness in diverse environments.

Extensive evaluations on the PASCAL VOC 2012 dataset demonstrate that our method outperforms mainstream algorithms in terms of precision and overall detection efficacy.

The paper is organized as follows: Section 2 reviews related work, Section 3 presents the chosen methodology, including an overview of YOLOv8 and the proposed SO-YOLOv8 network architecture, Section 4 analyzes the results, Section 5 discusses the findings, and Section 6 concludes the paper.

2. Related work

Small object detection has been a difficult problem in computer vision. Traditional methods like R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2016) are known for being highly accurate, but they come with downsides—mainly slower processing and a need for complex training. Earlier on, people relied on sliding window techniques, which not only took a lot of time but also needed heavy computational resources. Then YOLO (You Only Look Once) came along, changing things up by approaching object detection as a single-step regression task. This gave rise to single-stage detectors, such as the YOLO series (Du, 2018) and SSD (Liu et al., 2016), which only require one pass of a neural network to detect objects and classify them. This efficiency makes them perfect for real-time uses, but they often lose accuracy with smaller objects. Developments related to the YOLO series have been made in response to this trade-off. Liu, Wang, Zhou, Fu, Ma et al. (2020) revised the YOLOv3 architecture to improve the Resblock and introduce extra convolution operations to retain essential spatial details for detecting smaller objects. In 2020, Mixed YOLOv3-LITE (Zhao et al., 2020) introduced a streamlined object detection model, evaluated on the PASCAL VOC dataset, achieving a mean Average Precision (mAP) of 48.25%. Similarly, Wang, Xie, Zhang, Chen, Wen et al. (2021) worked on image super-resolution with an FTT (Feature Texture Transfer) module that sharpens image quality and lowers noise, boosting both speed and accuracy. Some new models focus on specific applications, too. YOLO-MXANet was designed for small objects in traffic scenes (He, Cheng, Zheng, & Wang, 2021), while YOLO-Fine is more suited for remote sensing (Pham, Courtrai, Friguet, Lefèvre, & Baussard, 2020). Zhang, Xia, Huang, Wang, and Akindele (2023) introduced ETAM, an encoder leveraging attention mechanisms to capture details with precision, and demonstrated substantial improvements in small object detection across various datasets, achieving a mean Average Precision (mAP) of up to 91.7 for small objects. A refined Fire-YOLO deep learning algorithm is introduced by Zhao, Zhi, Zhao, and Zheng (2022) to enhance the detection of small fire and smoke-like targets in forest fire imagery. YOLO-TLA (Ji, Yu, Gao, Wang, & Yuan, 2024) builds upon the YOLOv5 framework to balance computational efficiency with detection accuracy, making it highly suitable for real-time applications. An improved approach, MSS-YOLOv5 (He, Su, Wang, Yu, & Luo, 2023) enhances object detection by effectively balancing speed and accuracy. Niu, Cheng, Shi, and Fan (2023) introduced the lightweight object detection network YOLOv8-CGRNet that uses context guidance and deep residual learning to improve detection accuracy and efficiency. Beyond these specialized YOLO-based models, other research efforts have focused on various frameworks such as transformer-based architectures (Cheng et al., 2023), CNNs (Yang, Fan, Chu, Blasch, & Ling, 2019), and attention-driven models (Ju et al., 2021). The recent advancements in small object detection have focused on addressing challenges related to occlusion, low resolution, and scale variations. Models such as DETR (DEtection TRansformer) (Huang & Li, 2024) have been used for real-time detection of small objects, while hybrid CNN-Transformer architectures (Chen et al., 2019) improve

Table 1
Advancements in small object detection techniques.

Method	Main Contributions	Description
R-CNN, Fast R-CNN, Faster R-CNN (Girshick, 2015; Girshick et al., 2014; Ren et al., 2016)	High accuracy	Use region proposals and selective search to separate objects, therefore enabling exact localization. These techniques, while accurate, their limited resolution in suggested areas often causes restrictions with small object detection.
YOLO series (YOLOv1-v9) (Du, 2018)	Real-time efficiency	Uses single-step regression-based detection with real-time performance, yet historically has difficulty with small objects due to coarse-grained feature representation.
SSD (Liu et al., 2016)	Single-stage detection	Uses multi-scale feature maps to detect objects of varying sizes within a single forward pass. However, it still encounters difficulties detecting very small objects due to limited fine-grained feature capture.
Mixed YOLOv3-LITE (Zhao et al., 2020)	Lightweight	Integrates lightweight depthwise separable convolutions to reduce model complexity, making it suitable for real-time edge deployment. It specifically improves detection of small-scale objects through enhanced feature representation.
FTT Module (Wang et al., 2021)	Image enhancement	Uses Feature Texture Transfer (FTT) for image super-resolution, significantly improving detection quality and accuracy by enhancing fine details crucial for detecting small objects.
YOLO-MXANet (He et al., 2021)	Traffic detection	Optimized for detecting small objects in urban traffic scenes, addressing challenges like detecting distant or small vehicles and pedestrians through multi-scale context modeling and attention mechanisms.
YOLO-Fine (Pham et al., 2020)	Remote sensing	Enhances object detection in aerial and satellite images. It improves small-object detection by adaptive anchor boxes and tailored feature fusion techniques.
ETAM (Zhang et al., 2023)	Attention-based	Incorporates an attention mechanism in its encoder structure to selectively emphasize fine-grained details, greatly enhancing the detection accuracy for small objects in complex environments.
Fire-YOLO (Zhao et al., 2022)	Fire detection	Optimized for detecting fire and smoke in environmental monitoring, effectively addressing challenges related to scale variance and low contrast.
YOLO-TLA (Ji et al., 2024)	Efficient real-time	Improves real-time detection accuracy of small objects by carefully balancing model depth and width, combined with multi-scale feature extraction.
MSS-YOLOv5 (He et al., 2023)	Speed-accuracy tradeoff	Employs multi-scale spatial enhancement techniques, effectively addressing scale variation issues inherent to small object detection tasks.
YOLOv8-CGRNet (Niu et al., 2023)	Lightweight detection	Introduces context-guided residual learning, significantly enhancing feature representation capabilities and thus improving detection performance of small, densely packed objects.
Transformer-based (Cheng et al., 2023; Huang & Li, 2024)	Feature learning	Leverages self-attention mechanisms, allowing better feature aggregation and scale-invariant representations, crucial for accurately detecting small objects in complex scenes.
CNNs and Attention Models (Ju et al., 2021; Yang et al., 2019)	Diverse applications	Combine convolutional structures with spatial attention, refining feature extraction and specifically enhancing the detection of small, clustered, or occluded objects.

object detection in drone-captured images. Also, UAV-based traffic monitoring small object detection models (Sun, Dai, Zhang, Chang, & He, 2022) tackled real-world aerial imagery challenges by integrating scale-adaptive learning techniques. These methods have shown improvements in detection accuracy across diverse applications. However, challenges like scale variations and computational demands persist, motivating further research. Collectively, these advancements underscore the increasing focus on achieving higher accuracy and greater efficiency in small object detection. Table 1 summarizes related work.

While previous models have made strides in addressing various object detection challenges, SO-YOLOv8 differentiates itself by specifically optimizing small object detection through a unique combination of advanced data augmentation, feature fusion, and multi-scale training, which will benefit real-world applications.

3. Methodology

This research aims to enhance the detection accuracy of small objects within the PASCAL VOC 2012 dataset (Shetty, 2016) by refining and optimizing the YOLOv8x model. The modifications are carefully tailored to address the challenges of small object detection, including adjustments to feature extraction and integration techniques that allow the model to capture finer details effectively. In this section, we discuss the comprehensive experimental setup, the rationale behind choosing the PASCAL VOC 2012 dataset, and the process of dataset preparation to ensure robust training and evaluation. We further elaborate on the selection of YOLOv8x as the base model, along with the specific enhancements incorporated, such as an SE (Squeeze-and-Excitation) block and advanced data augmentation techniques that collectively contributed to achieving a precision of 1.0 for small object detection, as illustrated in Fig. 1.

The proposed SO-YOLOv8 incorporates various novel and strategic improvements into the original YOLOv8 architecture. The incorporation of Squeeze-and-Excitation (SE) blocks facilitates enhanced recalibration of feature maps, markedly augmenting detection sensitivity and precision for small objects by adaptively directing the model's attention towards salient regions. Furthermore, our careful selection and integration of sophisticated data augmentation techniques (MixUp, Copy-Paste, and Random Erasing) directly confront prevalent challenges in the detection of small-scale objects, including inadequate visual representation, occlusion, and intricate backgrounds. We augment these improvements by methodically using multi-scale training methodologies to adeptly address object scale variations and diligently conducting hyperparameter tweaking specifically designed to enhance performance for small objects.

3.1. Experimental setup

This experiment as shown in Table 2, was carried out on a high-performance compute node with a PowerEdge R750x server, equipped with high-performance components to handle intensive deep learning tasks. The processors are Dual Intel Gold 6338 processors, each operating at 2.0 GHz, providing substantial computational power for data processing and model training. The setup included 4 NVIDIA A100

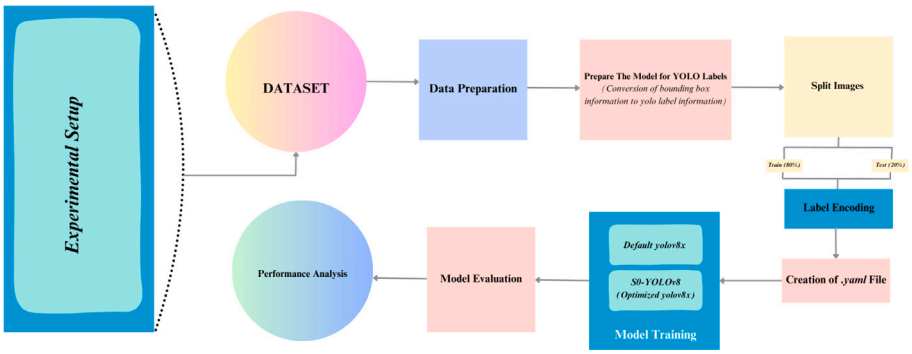


Fig. 1. Workflow from Experimental Setup to Performance Analysis.

Table 2
Experimental setup.

Component	Details
Server	PowerEdge R750x
Processors	Dual Intel Gold 6338, 2.0 GHz
GPUs	4 x NVIDIA A100, 80 GB each
RAM	1 TB
Operating System	Rocky Linux 8.10
Deep Learning Framework	PyTorch
Programming Language	Python 3.8
GPU Optimization	CUDA 11.8 and NVIDIA cuDNN

GPUs, each with 80 GB of memory, enabling efficient training and processing of large datasets. The server was equipped with 1 TB of RAM, ensuring sufficient memory for handling large data loads and optimizing model performance. Rocky Linux 8.10 was chosen as the operating system for its stability and compatibility with deep learning frameworks. The model was implemented using PyTorch, allowing flexibility in model customization and parameter tuning. The experiments were conducted using Python 3.8, ensuring compatibility with the deep learning libraries and tools. The additional libraries used are CUDA 11.8 for GPU acceleration, optimizing training speed and efficiency, and NVIDIA's cuDNN library, further enhancing GPU performance.

3.2. Dataset

Pascal VOC (Visual Object Classes) 2012 dataset (Shetty, 2016) was used for this study as it is a well-known benchmark for object detection tasks, mainly featuring medium and large-sized objects across various categories. The dataset was downloaded from the official PASCAL VOC 2012 website. Although PASCAL VOC is not specifically designed for small object detection, we intentionally chose it due to its low proportion of small objects, making it a challenging benchmark for evaluating SO-YOLOv8. Instead of using a dataset that inherently favors small object detection, our goal was to develop a model capable of detecting small objects even in scenarios where they are underrepresented. This approach tests SO-YOLOv8's ability to enhance small object detection performance in complex environments. The results demonstrate that SO-YOLOv8 effectively identifies small objects within PASCAL VOC, overcoming one of the key limitations of traditional YOLO models and validating its adaptability to datasets where small objects are more difficult to detect. The PASCAL VOC 2012 dataset consists of 11,540 images and 27,450 labeled objects, spanning 20 diverse object categories. Fig. 2 presents an exploratory analysis, including class distribution, object spatial location, size distribution, area distribution, attribute correlations, and sample training images. Specifically, small object categories primarily include distant pedestrians, bicycles, chairs, bottles, and potted plants. This relatively sparse representation of small objects presents additional challenges for detection models, making it a rigorous benchmark to evaluate the capability of SO-YOLOv8 to accurately detect and classify these less frequent, lower-resolution instances.

These insights substantiate the architectural improvements incorporated in the SO-YOLOv8 model, which are designed to tackle the challenges associated with the detection of small objects.

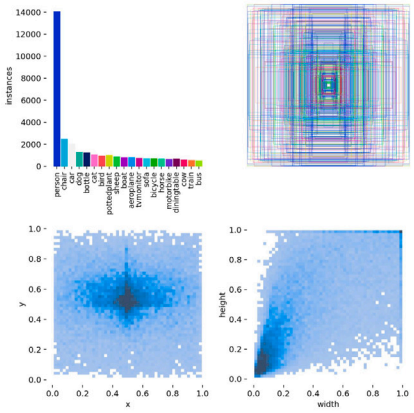
3.3. Overview of YOLOv8

YOLO was introduced in 2015 by Joseph Redmon (Redmon, 2016), since then it has been actively developed and refined by the computer vision community. Initially, YOLO was implemented in C using a custom deep learning framework called DarkNet (Setiyono, Amini, & Sulistyaningrum, 2021). This early version laid the groundwork for what would become a widely influential series of real-time object detection models, continuously evolving through community contributions and advancements. YOLOv8 (Jocher, Chaurasia, & Qiu, 2023) is one of the versions in the YOLO series of object detection models developed by the team at Ultralytics in January 2023, and designed to deliver high accuracy and speed in real-time detection tasks. This version builds on the strengths of its predecessors, incorporating innovative features and optimizations. These improvements make YOLOv8 well-suited for diverse object detection applications across multiple fields, where both performance and efficiency are important to consider. YOLOv8 is available in five variants categorized by the amount of parameters, *nano (n), small (s), medium (m), large (l), and extra large (x)*. We choose the YOLOv8x (Extra Large) variant as it has the capability to achieve superior accuracy, especially in scenarios requiring small object detection. YOLOv8x's extensive parameter set allows it to capture fine-grained details, making it ideal for achieving high precision in complex detection tasks.

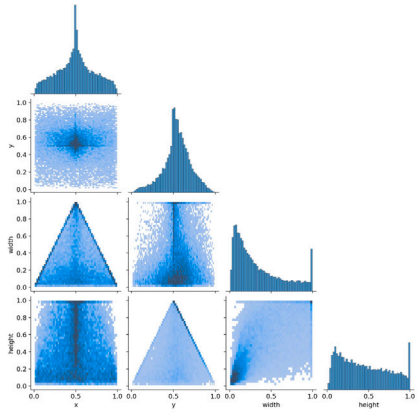
3.3.1. Architecture

The YOLOv8 architecture as shown in Fig. 3 can be broken down into three primary components:

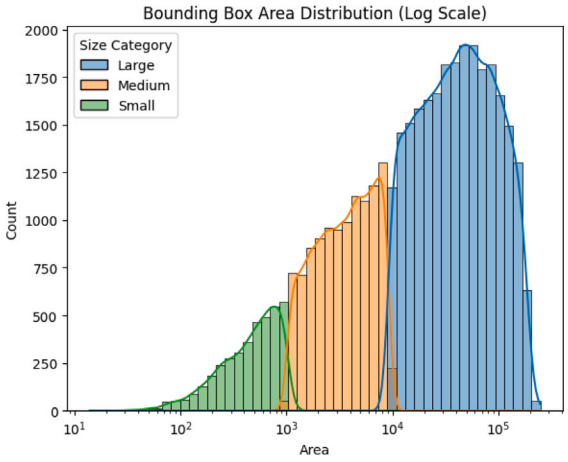
- *Backbone* — This part of the model is a convolutional neural network (CNN) that extracts important features from the input image. In YOLOv8, the backbone used for this task is a custom version of CSPDarknet53, which includes special cross-stage partial connections. These connections allow information to flow more smoothly between layers, ultimately helping the model achieve better accuracy.
- *Neck* — The neck, often called the feature aggregator, merges feature maps from different layers of the backbone. This allows the model to capture details at various scales. Instead of using the traditional Feature Pyramid Network (FPN), YOLOv8 introduces a new C2f module. This module combines high-level semantic information with low-level spatial details, which is especially useful for detecting small objects with greater accuracy.



(a) Class distribution and bounding box overview.

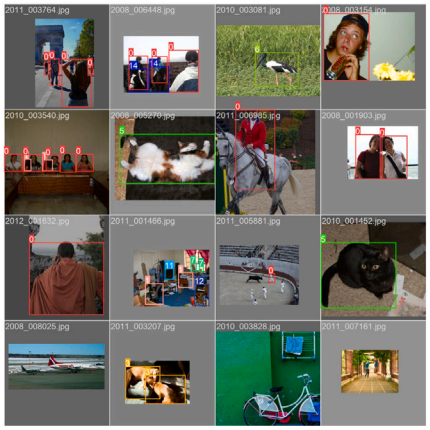


(b) Correlogram of object attribute relationships.



(c) Bounding box area distributions (log scale for area).

(d) Sample train batch images with bounding boxes.



(e) Train batch images showing object diversity.

Fig. 2. Visual Analysis of the PASCAL VOC 2012 Dataset, including class distribution, object relationships, sample train images, and bounding box size area distributions.

- *Head* — This part of the model is responsible for making the final predictions. In YOLOv8, the head includes multiple detection layers that generate bounding boxes, objectness scores, and class probabilities for each grid cell in the feature map. These outputs are then combined to create the final object detections.

3.4. Model configuration and customization

3.4.1. SO-YOLOv8

The proposed model SO-YOLOv8 was developed by selecting the YOLOv8x model due to its excellent feature extraction capabilities and efficiency in real-time object detection. As illustrated in Fig. 6, the SO-YOLOv8 model incorporates data augmentation techniques and architectural modifications to enhance small object detection. Squeeze-and-Excitation (SE) layers (Hu, Shen, & Sun, 2018) were integrated into the backbone to boost the sensitivity of small objects. The SE block, as depicted in Fig. 4, is a flexible computational module that can be applied to any transformation $F_{tr} : X \rightarrow U$, where $X \in \mathbb{R}^{H' \times W' \times C'}$ and $U \in \mathbb{R}^{H \times W \times C}$. For clarity, let us assume that F_{tr} represents a convolutional operation. Let $V = [v_1, v_2, \ldots, v_C]$ be a collection of learned filter kernels, with v_c representing the parameters of the c-th filter. The transformation output can be expressed as $U = [u_1, u_2, \ldots, u_C]$, where

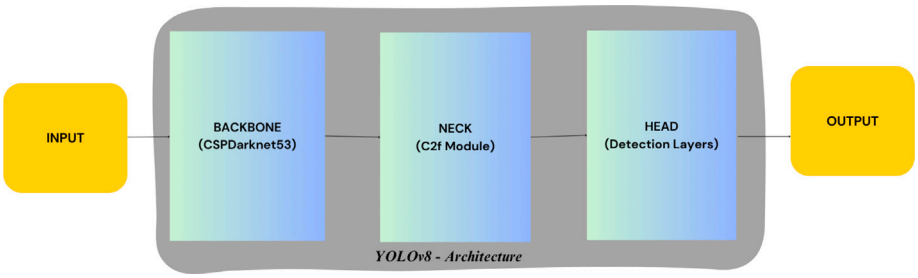


Fig. 3. Basic Architecture of YOLOv8.

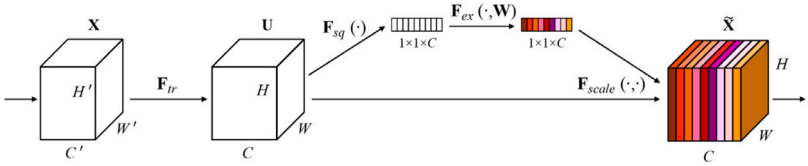


Fig. 4. Squeeze-and-Excitation Block (Hu et al., 2018).

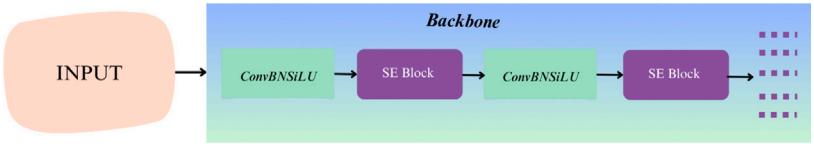


Fig. 5. Integration of SE Block with Backbone.

each u_c is defined as

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x_s \tag{1}$$

In this equation, $*$ represents the convolution operation, and $v_c = [v_c^1, v_c^2, \ldots, v_c^{C'}]$, where $X = [x_1, x_2, \ldots, x_{C'}]$. For simplicity, bias terms are not included. Each v_c^s is a 2D spatial kernel corresponding to one channel of v_c, and it processes the corresponding channel of X.

The output is a summation across all channels, meaning that channel dependencies are inherently embedded in v_c. However, these dependencies are intertwined with spatial correlations captured by the filters. The goal of the SE block is to increase the network's sensitivity to informative features and suppress less relevant ones. To achieve this, channel interdependencies are explicitly modeled, which recalibrates filter responses through two main steps: *Squeeze (global average pooling)* and *Excitation (adaptive scaling)*. The recalibrated responses are then fed into subsequent layers, enabling the network to focus more effectively on important features. In the YOLOv8 architecture, the SE block was incorporated right after the non-linearity that follows each convolutional layer as shown in Fig. 5. This strategic placement allows the SE block to adjust feature maps right after activation, improving the model's sensitivity to key features.

The neck's C2f module (Pan, Xu, Cheng, & Lian, 2024) was adjusted to improve multi-scale feature fusion (Yang, Wu, Du, & Zhang, 2021), and anchor boxes were recalibrated to better fit small object dimensions. Also, To boost the model's generalization across multiple object scales and views, we applied a range of data augmentation techniques (Kaur, Khehra, & Mavi, 2021), such as mosaic, random cropping, and color jittering. These augmentations were selected to help the model better handle diverse visual conditions and increase robustness during detection. The proposed SO-YOLOv8 model overview is shown in the Fig. 6. These changes were aimed at boosting precision and mAP, especially for detecting small objects, while keeping computational demands manageable.

3.4.2. Optimization and cost functions

The proposed SO-YOLOv8 model employs many cost functions to improve various aspects of detection of objects, ensuring precise classification, localization, and bounding-box regression. The loss functions adhere to the conventional YOLOv8 framework, however, the weighting parameters for these components were specifically modified to enhance small object detection. These weights were established empirically by executing several configurations and selecting the optimal values based on precision and mAP scores. The chosen values correspond with prior studies (Li, Chen, Wang, & Zhang, 2019; Lin, Goyal, Girshick, He, & Dollár, 2017b) in small object detection, wherein bounding box accuracy significantly influences detection quality more than classification in challenging scenarios. The loss function is composed of the following components:

- **Bounding Box Regression Loss**: The model uses a weighted loss for bounding box regression to ensure precise localization of small objects because of their limited pixel representation. This loss is primarily based on Intersection over Union (IoU) or its variants such as GIoU, DIoU, or CIoU, which provide better gradient updates when bounding boxes do not overlap perfectly. IoU Loss is given by:

$$\text{IoU Loss} = 1 - \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2}$$

  This loss encourages the model to maximize the overlap between predicted and ground truth bounding boxes. A high weight of 7.5 is assigned to this component, reflecting its importance in achieving accurate localization, especially for small objects (Tian, Zhao, Zhang, Yu, Sun et al., 2022), which are more susceptible to positional inaccuracies.
- **Classification Loss:** The classification loss, weighted at 0.5, penalizes incorrect predictions of object classes. This loss is typically based on Binary Cross-Entropy (BCE) loss, which is defined as:

$$\text{BCE Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \tag{3}$$

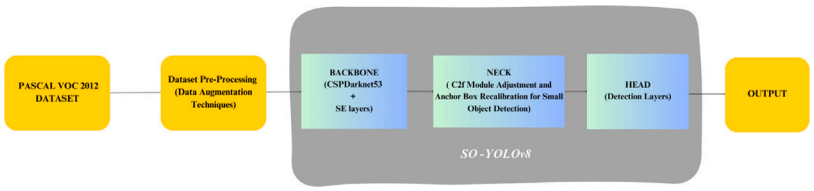


Fig. 6. Overview of the SO-YOLOv8 Model Configuration for Small Object Detection.

where y_i is the ground truth label, p_i is the predicted probability, and N is the total number of predictions. A weight of 0.5 has been assigned to the classification loss in order to mitigate the risk of imposing disproportionate penalties for misclassifications. Given that small objects typically possess fewer distinctive characteristics and present greater challenges in classification, an excessively high weight assigned to classification may result in instability during the learning process. This weight was selected to achieve an equilibrium between classification accuracy and localization efficacy (Lu, Chen, Wu, Tan, & Yu, 2023).

- **Distribution Focal Loss:** To address class imbalance and improve the detection of small objects, the model employs Distribution Focal Loss (DFL), which assigns higher weights to harder-to-classify samples. This component, weighted at 1.5, ensures that small objects, which are frequently more challenging to classify due to their low resolution (Lin et al., 2017b), receive sufficient attention.

$$\text{Focal Loss} = -\alpha(1 - p_t)^{\gamma} \log(p_t) \tag{4}$$

where p_t is the predicted probability for the ground truth class, α is a weighting factor, and γ is the focusing parameter that adjusts the loss contribution of well-classified examples. In SO-YOLOv8, γ is set to emphasize hard examples, particularly for small objects, improving overall detection performance.

By combining these loss components, bounding box regression, classification, and focal loss, the model achieves a balance between accurate localization, robust classification, and effective handling of small objects. The use of advanced optimization strategies, such as cosine learning rate scheduling and momentum, further ensures efficient convergence.

3.5. Hyperparameter configuration and optimization

In the SO-YOLOv8 model, designed for small object detection, we have implemented several critical adjustments to the model hyperparameters to optimize performance. We employed grid search for hyperparameter tuning. Grid search systematically evaluates all possible combinations within a predefined set of hyperparameters to identify the optimal configuration for small object detection. We tuned parameters such as learning rate, batch size, IoU threshold, and dropout rate, as detailed in Table 3. While grid search is computationally intensive, it ensures a thorough exploration of hyperparameter space and was preferred over alternative methods like Bayesian optimization and random search to achieve a comprehensive evaluation of model performance.

- Learning Rate {0.001, 0.005, 0.01} - Selected based on commonly used values in object detection research to balance model convergence and stability.
- Batch Size {8, 16, 32} - Chosen considering GPU memory constraints while ensuring efficient model training.
- IoU Threshold {0.5, 0.7, 0.75} - Tuned to balance localization precision and recall.
- Dropout Rate {0.1, 0.3, 0.5} - Adjusted to prevent overfitting while maintaining feature retention.

Table 3
Configuration changes and their impact from YOLOv8 to SO-YOLOv8.

Parameter	YOLOv8	SO-YOLOv8	Impact
Batch size	16	8	Smaller batch size enhances generalization but increases training time and reduces stability.
Learning rate	0.01	0.005	Lower learning rate provides refined updates, reducing overshooting risks.
Patience	50	20	Reduced patience speeds convergence, with a higher risk of early stopping.
Dropout	0.0	0.3	Dropout at 0.3 helps prevent overfitting with stronger regularization.
Data Augmentation	None	mixup=0.1, copy-paste=0.1, erasing=0.4	Augmentation increases data diversity, improving robustness and generalization.
Conf Threshold	Not specified	0.1	Lower threshold boosts recall but may introduce more false positives.
IoU Threshold	0.7	0.75	Higher IoU enforces stricter criteria, potentially increasing precision.
Multi-Scale Training	Not specified	Enabled	Improves generalization across sizes, benefiting small object detection.
Close Mosaic	10	10	Affects spatial learning by controlling image combination during training.
Cosine LR scheduler	Not specified	Enabled	Gradual LR reduction aids convergence in later stages.

After conducting thorough experiments, the Table 3 outlines configuration that yielded the best results. This structured and systematic tuning process ensured that SO-YOLOv8 achieved an optimal balance between precision and recall, significantly improving its ability to detect small objects while mitigating the risk of overfitting.

Each adjustment is specifically aimed at enhancing the model's ability to accurately detect small objects across varied conditions, ultimately contributing to improved precision and robustness in detection tasks.

- *Batch Size* — We reduced the batch size from 16 to 8, which helped improve the model's ability to generalize. While this smaller batch size did slow down training a bit, it allowed the model to make more fine-tuned updates, which is especially useful for detecting small objects where precision is key.
- *Learning Rate* — The learning rate was lowered from 0.01 to 0.005, with a cosine annealing schedule. By starting lower and gradually decreasing the rate, the model was able to make more refined adjustments during training, which reduced the risk of overshooting the optimal point. This gradual learning rate adjustment contributed to smoother convergence.
- *Patience* — To speed up convergence, we have reduced the patience parameter for early stopping from 50 epochs to 20. This helped prevent the model from lingering too long if it was not making significant improvements. However, it did carry a risk of

stopping too soon, so we had to balance this carefully to ensure we were not cutting off training before reaching good results.

- *Dropout* — To help prevent overfitting, we added dropout with a rate of 0.3. Dropout adds a layer of randomness during training by temporarily "turning off" certain neurons, which encourages the model to rely less on specific paths and more on general features. This regularization step was especially helpful for complex tasks like small object detection, where the model needs to be adaptable.
- *Data Augmentation* — We applied several data augmentation techniques, like MixUp, Copy-Paste, and random erasing. These methods essentially "create" new training examples by altering existing ones, which helped the model become more resilient and better at generalizing. For example, MixUp blends two images together, which gives the model a broader view of object variations, while Copy-Paste lets it see objects in new contexts.
- *Confidence Threshold* — We set a relatively low confidence threshold of 0.1, which allowed the model to pick up on a wider range of possible detections. This helped increase recall, though it did introduce more potential false positives. I managed this trade-off by tweaking other parameters to maintain precision.
- *IoU Threshold* — The Intersection over Union (IoU) threshold was set at 0.75 to apply stricter criteria for detections. This ensured that only the most accurate predictions were counted, which is important when working with small objects that might easily blend into the background or be mistaken for noise.
- *Multi-Scale Training* — Multi-scale training was enabled to help the model detect objects at various sizes, focusing on improving small object detection. By training the model on images at different scales, it could generalize better across a range of object sizes, from very small to large.
- *Close Mosaic and Cosine Learning Rate Scheduler* — The close mosaic setting remained consistent, helping the model learn spatial relationships by combining images in unique ways. The cosine learning rate scheduler helped reduce the learning rate gradually, allowing the model to settle into a more stable state as training progressed.

These hyperparameter tweaks were essential in optimizing the SO-YOLOv8 model for small object detection, allowing it to achieve a balance between accuracy and efficiency. The results, as we see, reflect the impact of these carefully chosen configurations. Also, the sensitivity analysis indicates that the performance boost is the result of architectural innovations such as SE Blocks rather than hyperparameter optimization.

4. Results & analysis

The performance evaluation of the proposed SO-YOLOv8 model in comparison to the baseline YOLOv8 model reveals that SO-YOLOv8 achieves superior results across multiple key metrics. Specifically, we present our findings using metrics such as the confusion matrix, precision, mean Average Precision (mAP), and recall, each highlighting the advancements in detection accuracy and reliability offered by SO-YOLOv8. Also, visual results are provided to show the model's enhanced capability in accurately detecting small objects, further demonstrating its effectiveness over the original YOLOv8 model.

4.1. Confusion matrix

The confusion matrix shows the true positives, false positives, and false negatives, giving insight into the model's strengths and weaknesses for each class. The confusion matrix *(shown in Fig. 7)* demonstrates SO-YOLOv8's improved performance with high true positive rates across most classes in comparison with the confusion matrix for YOLOv8. The diagonal values in the confusion matrix for SO-YOLOv8 are consistently elevated across the majority of classes, signifying enhanced true positive rates and overall accuracy. However, YOLOv8 exhibits lower diagonal values, indicating comparatively poorer detection performance. Also, SO-YOLOv8 is able to successfully resolve concerns related to misclassification. When compared to YOLOv8, the off-diagonal values in its confusion matrix are significantly lower, which highlights the reduction in the number of false positives and false negatives detected by the proposed model. This advantage is particularly apparent in complex environments that involve cluttered backgrounds, objects that overlap one another, or objects that are extremely small in size. The confusion matrix study demonstrates that the proposed improvements are effective and establishes SO-YOLOv8 as a considerable improvement over the baseline YOLOv8 model. Table 4 further provides a clear quantitative comparison of the true positive rates (TPRs) for each class, demonstrating how SO-YOLOv8 consistently improves detection across individual categories compared to the baseline YOLOv8.

Table 4
Quantitative comparison of true positive rates per class (YOLOv8 vs. SO-YOLOv8)

Class	YOLOv8 (TPR)	SO-YOLOv8 (TPR)
Person	0.78	**0.94**
Chair	0.55	**0.79**
Car	0.72	**0.87**
Dog	0.76	**0.81**
Bottle	0.52	**0.74**
Cat	0.83	**0.90**
Bird	**0.86**	0.81
Potted Plant	0.59	**0.87**
Sheep	**0.78**	0.72
Boat	0.49	**0.86**
Aeroplane	0.72	**0.90**
TV monitor	0.73	**0.96**
Sofa	**0.71**	0.69
Bicycle	0.70	**0.84**
Horse	0.72	**0.83**
Motorbike	0.86	**0.93**
Dining Table	0.62	**0.76**
Cow	0.68	**0.82**
Train	0.87	**0.92**
Bus	0.76	0.76

4.2. Precision, recall and mAP

Precision (Ranawana & Palade, 2006) measures the proportion of true positive detections to the total number of positive predictions, true positive (TP) and false positives (FP).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

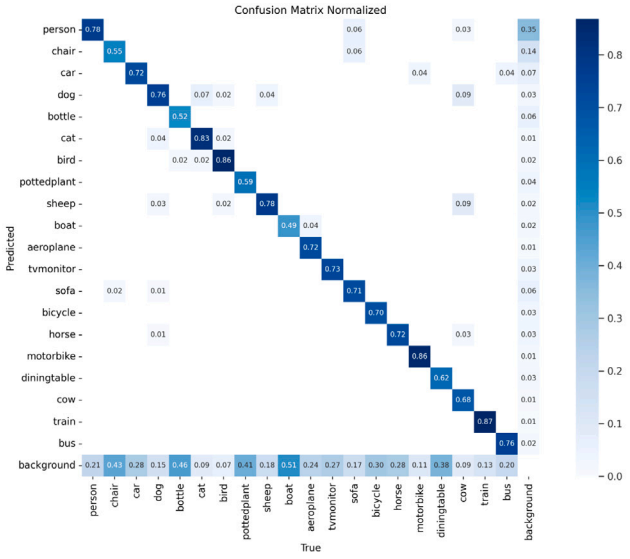
Recall (Lavie, Sagae, & Jayaraman, 2004) represents the proportion of true positive detections to the total number of actual positives, true positives(TP) and false negatives(FN).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

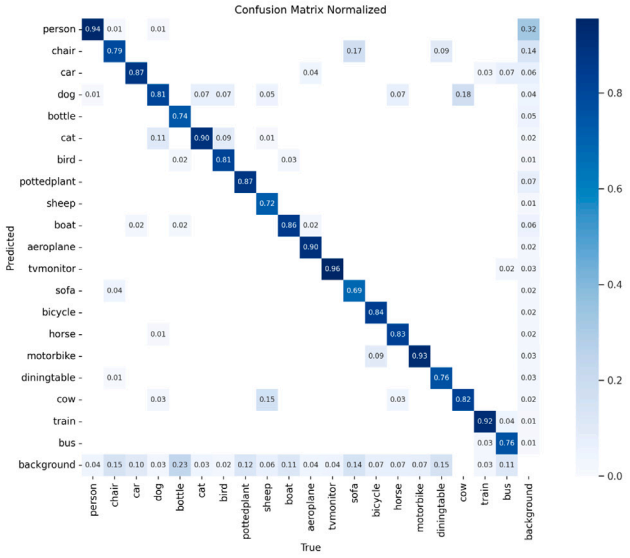
The mAP (Henderson & Ferrari, 2017) is the average precision across all classes (N) and is calculated as the mean of Average Precision (AP) values. AP is typically calculated as the area under the precision–recall curve for each class.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{7}$$

The SO-YOLOv8 model also outperforms YOLOv8 in terms of precision, recall and mAP(@0.5) which is reflected in its enhanced detection accuracy and reduced error rates for challenging small objects. This improvement can be attributed to the model's optimized architecture. The precision images as shown in Fig. 8 highlight the differences in object detection accuracy between YOLOv8 and SO-YOLOv8.



(a) Confusion Matrix - YOLOv8



(b) Confusion Matrix of SO-YOLOv8

Fig. 7. Comparison of Confusion Matrix - YOLOv8 and SO-YOLOv8 (*SO-YOLOv8 shows a higher true positive rate for objects compared to YOLOv8, leading to improved detection accuracy and fewer false negatives.*)

Table 5
Performance Comparison of YOLOv8 and proposed SO-YOLOv8. (*The **6% increase in precision** and **1% improvement in mAP** are highlighted.*)

Model	Precision	Recall	mAP@0.5
YOLOv8	0.94	0.89	0.78
SO-YOLOv8 *(proposed)*	**1.00 (+6%)**	**0.89**	**0.79 (+1%)**

While recall values remained consistent across YOLOv8 and SO-YOLOv8, significant improvements were observed primarily in precision (reduction in false positives), justifying our emphasis on the metrics. The negligible difference in recall indicates both models similarly capture most relevant objects, but SO-YOLOv8 does so with greater reliability.

The Table 5 and Fig. 9 present the results of SO-YOLOv8's performance, along with a comparison to the baseline YOLOv8 model.

The SO-YOLOv8 model achieved a precision of 1.0, marking a significant improvement over YOLOv8 by reducing false positives and enhancing detection accuracy for small objects. This enhancement demonstrates the model's robustness, especially in scenarios with complex backgrounds and varying object scales. Also, the improved recall and mAP scores reflect the model's enhanced ability to detect small objects comprehensively and accurately.

The Fig. 10 shows the performance of the cost function (training and validation loss curves). This figure illustrates the model's convergence and stability during training.

To further illustrate the effectiveness of SO-YOLOv8 in detecting small objects, we provide a series of visual examples as shown in Fig. 11 that show its performance across various challenging object detection tasks. In each case, we present side-by-side comparisons with YOLOv8 to clearly highlight the improvements offered by our proposed model. Through these comparisons, we emphasize the advancements in precision, recall, and mAP achieved by SO-YOLOv8, further highlighting its robustness and effectiveness in real-world small object detection scenarios. SO-YOLOv8 demonstrates significant improvements in detecting small and occluded objects compared to YOLOv8. As shown in Fig. 11, SO-YOLOv8 correctly identifies small objects that YOLOv8 fails to detect.

4.3. Class-wise performance analysis

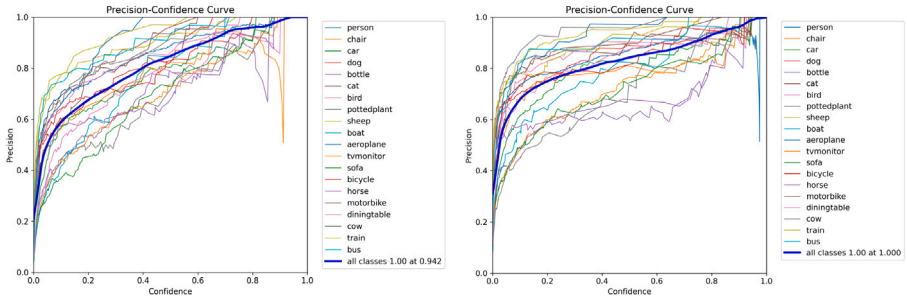
To further validate the effectiveness of SO-YOLOv8, we performed a class-wise analysis using True Positive Rates (TPR) (refer to Table 4 and Fig. 7). The analysis verified that, in most small object classes, SO-YOLOv8 significantly improves detection accuracy as compared to the baseline YOLOv8. The most significant improvements were observed in the most challenging classes, including "chair" (from 0.55 to 0.79), "bottle" (from 0.52 to 0.74), "boat" (from 0.49 to 0.86), and "TV monitor" (from 0.73 to 0.96). Although few object classes defined by complex textures or high background similarity ("sheep", "sofa", "bird") remain challenging due to inherent detection difficulties at low resolutions, SO-YOLOv8 still outperforms the baseline YOLOv8 model across almost all evaluated classes. This confirms the robustness and balanced performance of the proposed architectural enhancements.

4.4. Qualitative analysis of SO-YOLOv8 improvements over YOLOv8

We have conducted a qualitative analysis on challenging small object detection cases, including occlusion, low contrast, and cluttered backgrounds. Fig. 12 shows comparisons between YOLOv8 and SO-YOLOv8, emphasizing the substantial enhancements obtained by our proposed model. SO-YOLOv8 significantly enhances detection in three key areas, small object visibility, handling occlusion and crowded scenes. YOLOv8 often fails to detect small and distant objects, especially when they blend into the background, such as pedestrians or vehicles appearing far away.

However, SO-YOLOv8 successfully detects these objects. Fig. 12(a) and 12(d) shows the original images, providing reference for comparison, We observe that YOLOv8, (Fig. 12(b)) missed the distant car (small object) in the background. In contrast, SO-YOLOv8 (Fig. 12(c)) successfully improves detection by detecting all the objects, demonstrating its better feature extraction and recalibration capabilities. This improvement is primarily due to the incorporation of the SE block and enhanced feature recalibration, which enable the model to extract finer details, thereby boosting small object detection performance. Similarly, Fig. 12(e) shows a densely populated scene where again YOLOv8 has failed to detect certain individuals but, SO-YOLOv8 uses feature recalibration and mixup augmentation, which refine the model's ability to distinguish objects even in dense environments. As a result, it achieves higher detection accuracy in high-density scenarios as shown in Fig. 12(f), making it more reliable in complex environments.

These qualitative comparisons confirm the success of our changes incorporating multi-scale data augmentation methods and SE block for better feature recalibration. SO-YOLOv8 beats YOLOv8 in real-world



(a) Precision - YOLOv8 (b) Precision - SO-YOLOv8

Fig. 8. Precision comparison between YOLOv8 and SO-YOLOv8. (*SO-YOLOv8 achieves a precision of 1.0, improving from 0.94 in YOLOv8, highlighting the effectiveness of the proposed enhancements.*)

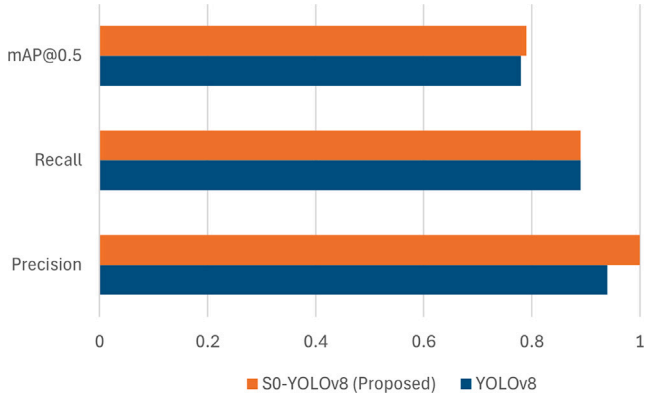


Fig. 9. Performance Comparison of YOLOv8 and proposed SO-YOLOv8. (*SO-YOLOv8 improves small object detection by reducing false positives while maintaining recall).*

small object detection scenarios by tackling occlusion problems, background noise, and very small object misdetections, therefore making it a better fit for applications in surveillance, autonomous navigation, and medical imaging.

4.5. Limitations of SO-YOLOv8

SO-YOLOv8 is highly optimized for small object detection and has shown strong performance, outperforming baseline models. However, some difficult situations exist as with every deep learning-based detection system, which offers chances for ongoing development. Fig. 13 shows the failure cases of SO-YOLOv8. The following limitations have been identified:

- Low-Contrast Areas or Crowded scene: Objects with colors like their background such as white uniformed athletes on a brilliant field may be difficult to tell apart. Techniques like contrast-aware augmentation might help to reduce this. Also, SO-YOLOv8's performance may be challenged in extreme high-density scenarios whereby objects are greatly overlapping or merging into background noise. For example, detecting people standing shoulder-to-shoulder in very crowded areas might result in bounding box overlaps.
- Extreme Occlusion or very distant objects: SO-YOLOv8 recognizes occluded items rather well, hence cases of severe occlusion that is, pedestrians entirely concealed by other objects may cause lower confidence detections or missed objects. Also, SO-YOLOv8 struggles with ultra-tiny objects (e.g., distant birds, far-away pedestrians, small street signs) due to their minimal pixel representation.
- High Aspect Ratio Objects: Thin or elongated objects, such cables or distant streetlights, may not always be effectively identified depending on the adaption limits of the receptive field.
- Low-Light and Nighttime Conditions: Reduced object visibility under quite bad lighting reduces detection confidence.

4.6. Quantitative comparison with state-of-the-art models

To critically assess the strengths and limitations of SO-YOLOv8 compared to state-of-the-art (SOTA) object detection models, we evaluate its performance in various aspects, including detection accuracy, computational complexity, uncertainty handling, and feature representation strategies as shown in Table 6. Many existing models struggle with small object detection due to issues such as loss of fine-grained details, high sensitivity to occlusion, and challenges in uncertainty modeling. we have structured our comparative analysis by following the methodology used in recent works (Hemmati & Rahmani, 2024; Hemmati & Zarei, 2024) , ensuring a standardized evaluation.

Table 6 demonstrates that SO-YOLOv8 achieves the highest precision (1.00) and mAP (0.79), while maintaining competitive recall (0.89). Traditional object detection models, including SSD and Faster R-CNN, are restricted in their ability to detect small objects due to their dependence on fixed feature representations. Conversely, SO-YOLOv8 employs a SE block and multi-scale training to dynamically recalibrate features, thereby improving the accuracy of detection in complex environments and substantially improving small object detection. Also, conventional models encounter difficulties in managing feature uncertainty and occlusion, which results in diminished performance in real-world scenarios. SO-YOLOv8 addresses these challenges by integrating sophisticated data augmentation techniques, which enhance its adaptability and robustness in unpredictable and congested environments. Although SO-YOLOv8 introduces a minor increase in computational cost, this trade-off is mitigated by substantial improvements in precision and mAP, which guarantee its applicability for real-world applications that necessitate high detection reliability, such as autonomous systems and medical imaging. The comparative analysis we have conducted by incorporating these findings offers a more critical assessment of the current limitations and emphasizes the effective way in which SO-YOLOv8 addresses critical obstacles in the detection of small objects.

4.7. Cross-dataset generalization on unseen small object datasets

To access the generalization capability of SO-YOLOv8 beyond the PASCAL VOC dataset, We evaluated SO-YOLOv8 on VisDrone and TinyPerson, two datasets especially intended for small object detection.

- VisDrone - Aerial images containing small vehicles and pedestrians with extreme scale variations.

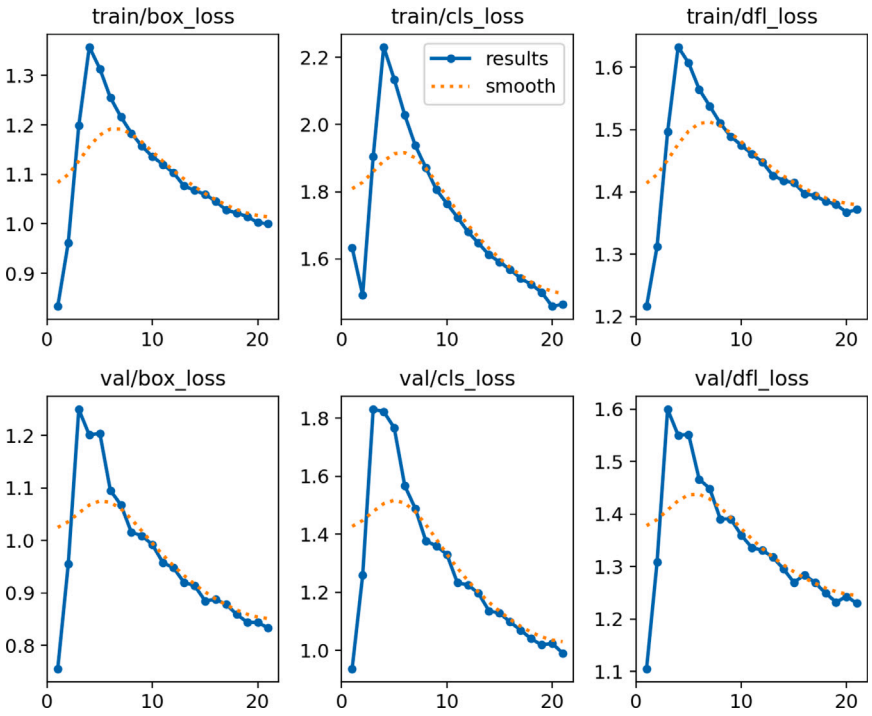


Fig. 10. Loss Curves for Training and Validation: Box Loss, Classification Loss, and DFL Loss over 20 epochs. The top row represents training losses, while the bottom row represents validation losses. The smooth validation loss curve indicates stable convergence of SO-YOLOv8, demonstrating improved generalization over YOLOv8.

Table 6
Performance comparison of SO-YOLOv8 with other SOTA models on pascal VOC 2012.

Model	Precision	Recall	**mAP@0.5**	Complexity	Uncertainty handling	Limitations
Fast R-CNN (Girshick, 2015)	–	–	0.68	High	Poor	Not optimized for real-time detection
Faster R-CNN (Ren et al., 2016)	–	–	0.70	High	Poor	Slow inference, region proposal overhead
SSD (Liu et al., 2016)	–	–	0.74	Moderate	Limited	Struggles with small object detection
YOLO (Redmon, 2016)	–	–	0.57	Low	Poor	Lacks robustness in uncertain environments
YOLOv3 (Zhao et al., 2020)	0.42	0.68	0.55	Low	Poor	Weak feature extraction for small objects
Mixed YOLOv3-LITE (Zhao et al., 2020)	0.49	0.69	0.48	Low	Moderate	Reduced accuracy compared to full-scale models
Few-shot object detection (Xiao, Lepetit, & Marlet, 2022)	–	–	0.57	High	Limited	Performance drops on small objects
YOLOv7 (Lou et al., 2023)	–	–	0.69	Moderate	Limited	l struggles with detecting small objects
DETR (Huang & Li, 2024)	–	–	0.78	High	Limited	Slow convergence, struggles with small objects
SO-YOLOv8 *(proposed)*	**1.00**	**0.89**	**0.79**	Moderate	Improved	Slightly higher computational cost due to SE block

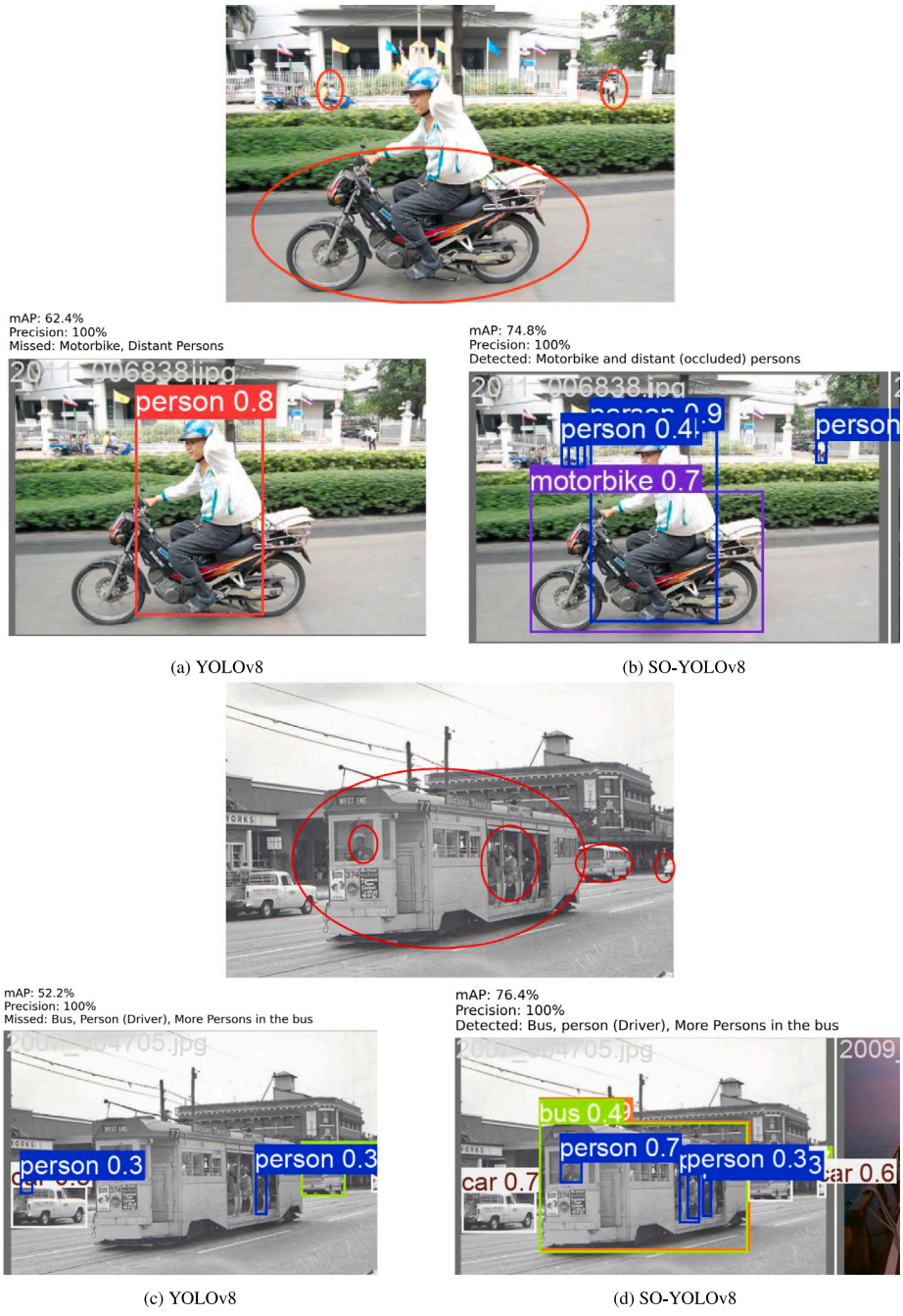
Table 7
Performance comparison of SO-YOLOv8, YOLOv8 and other state-of-art detection models on VisDrone and TinyPerson datasets.

Dataset	Model	Precision	Recall	mAP@0.5
VisDrone	CornerNet (Law & Deng, 2018)	–	–	0.31
	Light-RCNN (Li, Peng, Yu, Zhang, Deng et al., 2017)	–	–	0.33
	FPN (Lin, Dollár, Girshick, He, Hariharan et al., 2017a)	–	–	0.32
	Cascade-RCNN (Cai & Vasconcelos, 2018)	–	–	0.31
	YOLOv8 *(baseline)*	0.92	0.63	0.47
	SO-YOLOv8 *(proposed)*	**0.97**	**0.69**	**0.51**
TinyPerson	RetinaNet (Yu, Gong, Jiang, Ye, & Han, 2020)	–	–	0.43
	Faster RCNN-FPN (Lin et al., 2017a)	–	–	0.47
	Swin-T (Liu et al., 2021)	–	–	0.40
	YOLOv8 *(baseline)*	0.76	0.59	0.48
	SO-YOLOv8 *(proposed)*	**0.88**	**0.61**	**0.49**

- TinyPerson - Crowded urban scenes with highly occluded small pedestrians.

As seen in Table 7, SO-YOLOv8 achieves higher performance compared to YOLOv8 and other state-of-art models on both datasets. Fig. 14 compares the performance of YOLOv8 and SO-YOLOv8 on VisDrone Dataset. Standard YOLOv8 fails to detect many distant and small objects, while SO-YOLOv8 improves detection, especially for pedestrians and vehicles. This highlights the advantage of our feature recalibration strategy for aerial imagery, reducing false negatives in urban environments.

Fig. 15 compares the performance of YOLOv8 and SO-YOLOv8 on the TinyPerson dataset. YOLOv8 struggles with detecting small, distant pedestrians, often misclassifying them as background or unrelated objects. In contrast, SO-YOLOv8 improves recall for small human figures, leveraging feature recalibration to enhance small-scale feature recognition. This demonstrates the effectiveness of SO-YOLOv8 in crowded environments where small object visibility is crucial.



(a) YOLOv8 (b) SO-YOLOv8

(c) YOLOv8 (d) SO-YOLOv8

Fig. 11. Comparison of YOLOv8 and SO-YOLOv8 Detection Results *(Red encircled objects show missed detections by YOLOv8 but detected by SO-YOLOv8).*

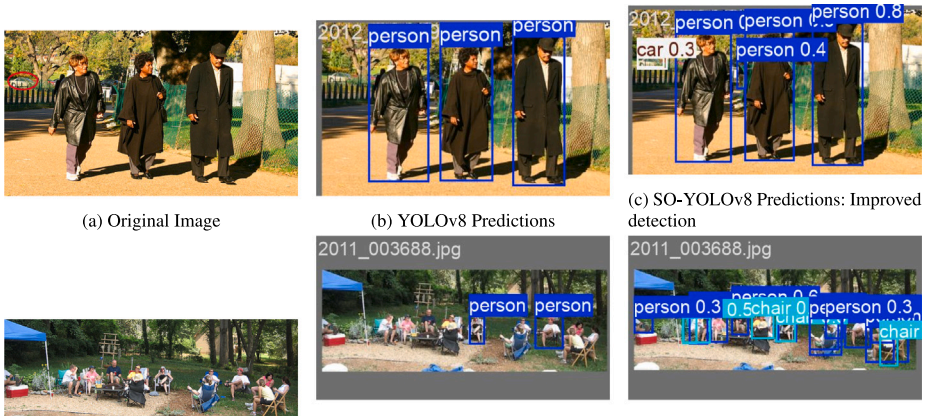
4.8. Computational complexity analysis

The computational complexity of SO-YOLOv8, in comparison to YOLOv8 model is evaluated in terms of floating point operations (FLOPs), total parameters, and inference time per image to evaluate the impact of the added Squeeze-and-Excitation (SE) block on computational efficiency. The SE block's integration results in a slight increase in computational complexity, but it substantially improves the detection of small objects.

The analysis in Table 8 indicates that the FLOPs and parameter count have increased by approximately 2%, which is a reasonable trade-off given the 6% boost in precision and 1% improvement in mAP. Although the SE block contributes to this increase, its impact on inference time remains minimal. This is likely due to its ability to enhance feature selection and reduce unnecessary computations, leading to a more efficient processing pipeline. In practical applications like traffic monitoring and medical imaging, ensuring real-time performance is essential. SO-YOLOv8 achieves an inference speed of 4.3 ms per image, which is actually faster than YOLOv8 (4.8 ms per image), despite the minor computational overhead, making it well-suited for real-world deployment. To further enhance efficiency on edge devices, techniques such as model pruning, quantization, and

Table 8
Comparison of Computational Complexity of YOLOv8 and SO-YOLOv8 *(proposed)*

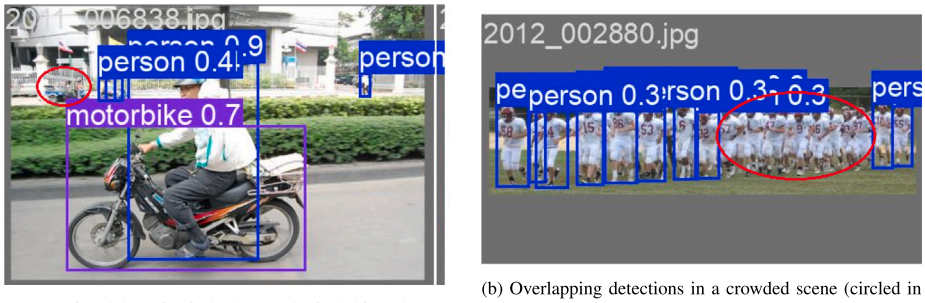
Model	Layers	Params (M)	FLOPs (B)	Infer time (ms/img)	Train time (s/epoch)
YOLOv8x	249	68.2	257.8	4.8	213.4
SO-YOLOv8 *(proposed)*	276	69.8	263.0	4.3	258.6



(a) Original Image (b) YOLOv8 Predictions (c) SO-YOLOv8 Predictions: Improved detection

(d) Original Image (e) YOLOv8 Predictions: Multiple missed objects (f) SO-YOLOv8 Predictions: Improved detection of people

Fig. 12. Qualitative comparison between YOLOv8 and SO-YOLOv8.



(a) Missed detection in background (circled in red). (b) Overlapping detections in a crowded scene (circled in red).

Fig. 13. Failure Cases of SO-YOLOv8. (a) The model fails to detect a small object due to background blending. (b) Detection errors in high-density environments where objects overlap significantly.

(a) YOLOv8 Prediction (b) SO-YOLOv8 Prediction

Fig. 14. Comparison of object detection results on the VisDrone dataset. SO-YOLOv8 improves small object detection, reducing false negatives in aerial scenes.

(a) YOLOv8 Prediction (b) SO-YOLOv8 Prediction

Fig. 15. Comparison of object detection results on the TinyPerson dataset. SO-YOLOv8 improves small object detection, reducing false negatives in crowded scenes.

Table 9
Sensitivity analysis of key hyperparameters.

Models	mAP	Precision	Recall
YOLOv8	0.76	0.97	0.85
SO-YOLOv8 (*proposed*)	0.79	1.0	0.89

knowledge distillation can be explored. These methods would help reduce computational overhead while preserving detection accuracy, allowing SO-YOLOv8 to operate efficiently on low-power AI systems without compromising performance.

4.9. Sensitivity analysis

To explicitly isolate the effect of SE Blocks, we ensure that the performance boost is not just due to hyper-parameter adjustments. We trained both YOLOv8 and SO-YOLOv8 (*proposed*) using identical hyper-parameter settings (learning rate - 0.005, Dropout - 0.3, IoU Threshold - 0.75) as shown in Table 9. The sensitivity analysis result's confirm that SO-YOLOv8 consistently outperforms YOLOv8 and even with constant Hyper-parameter tuning, proving that the observed improvements stem from our architectural modifications (*incorporation of SE block*) rather than fine tuning.

Also, the visual results of YOLOv8 struggle to detect small objects in the image, whereas SO-YOLOv8 (proposed) demonstrates enhanced detection capabilities for these challenging instances. This emphasizes the effectiveness of the SE Blocks in addressing the limitations of YOLOv8, particularly in small object detection.

4.10. Ablation study

To quantify the contribution of each architectural modification and training strategy to the overall performance of SO-YOLOv8, we conducted an ablation study shown in Table 10, quantifying computational trade-offs (FLOPs, parameters, and inference time) and highlights the individual contributions of each architectural and augmentation component. The SE block provides significant accuracy improvements (+1% mAP, +3% precision) with minimal computational overhead (+1.2% FLOPs increase). Each augmentation technique independently contributes to generalization, with MixUp, Copy-Paste, and Random Erasing showing incremental precision improvements without notably affecting inference speed. This detailed analysis supports informed decisions regarding the deployment of SO-YOLOv8 in computationally sensitive real-world scenarios.

While the mAP and recall improvements from individual modifications may seem minor, their impact on real-world detection robustness is significant. The addition of the SE Block enhances feature recalibration, improving small object visibility, especially in low-contrast and occluded environments. The SE block clearly improves precision by 3%, while increasing FLOPs by 1.2%, therefore illustrating the trade-off between computing efficiency and detection accuracy. Data augmentation methods like MixUp, Copy-Paste, and Random Erasing help to further improve model generalization and resilience. Random Erasing forces the model to acquire robust feature representations, Copy-Paste helps detect occluded objects, and Mix-Up augments feature blending. Every augmentation contributes uniquely to the final performance of SO-YOLOv8, with MixUp improving recall and Copy-Paste increasing small object visibility. Hyperparameter optimization refines the final model, stabilizing precision at 1.00 and mAP at 0.79. Although these improvements introduce a minor computational increase, SO-YOLOv8 maintains a faster inference time (4.3 ms per image) than YOLOv8 (4.8 ms per image). This validates that the modifications do not significantly hinder real-time applicability, making SO-YOLOv8 a practical solution for edge devices.

5. Discussion

The findings show that SO-YOLOv8 markedly surpasses the baseline YOLOv8 model, especially in the detection of small objects. While SO-YOLOv8 demonstrates a relatively small rise in mean Average Precision (mAP) from 0.78 to 0.79, the more significant metric is the 6% enhancement in precision, which rises from 0.94 to 1.0. This is because small objects contribute minimally to the overall mAP due to their infrequent occurrence in datasets such as Pascal VOC. However, precision directly impacts real-world applications where false positives can have severe consequences, such as in autonomous driving, medical imaging, and security surveillance. The significant improvements in SO-YOLOv8 result from critical architectural modifications that substantially boost small object detection performance. The incorporation of Squeeze-and-Excitation (SE) blocks recalibrates feature maps by dynamically adjusting the importance of each channel, allowing the network to focus more on small object features that might otherwise be suppressed by dominant larger objects. This recalibration also ensures that small objects are detected with greater accuracy, thereby enhancing the overall precision observed. The comprehensive analysis in Section 4.3 shows that SO-YOLOv8 consistently achieves greater detection performance across most object categories, demonstrating its strong overall effectiveness. The model achieves substantial improvements even on object classes which were previously challenging for YOLOv8. While certain inherently difficult small object classes – such as those involving high occlusion or complex textures – still present some detection challenges, SO-YOLOv8 significantly reduces these issues compared to baseline YOLOv8. The ablation study confirms that the SE block is the most influential modification in SO-YOLOv8, significantly improving small object detection accuracy while maintaining computational efficiency. The SE block contributed the most to improving small object detection, while data augmentation and hyperparameter tuning refined precision and model stability. This confirms that SO-YOLOv8's performance improvements stem from both architectural enhancements and optimized training strategies. Also, multi-scale training improves the model's capacity to generalize across different object sizes, mitigating the difficulties associated with scale differences and enhancing recall. Moreover, sophisticated data augmentation methods, like MixUp, Copy-Paste, and Random Erasing, enhance the diversity of the training dataset, hence fortifying the model's resilience against occlusions and intricate backdrops. These results confirm the resilience and versatility of SO-YOLOv8's design. The research demonstrates that SO-YOLOv8 is an effective model for detecting small objects, exhibiting improvements in accuracy. The integration of the SE module and multi-scale training, along with sophisticated data augmentation methods, reveal the efficacy of architectural improvements in tackling small object detection issues.

6. Conclusion & future scope

This study presents SO-YOLOv8, an optimized object detection model that significantly enhances the baseline YOLOv8, particularly in detecting small objects. By incorporating advanced data augmentation techniques, feature fusion strategies, and multi-scale training, SO-YOLOv8 achieves higher precision and reduces false positives compared to the original model. The experimental results demonstrate that SO-YOLOv8 achieves a 1% improvement in mAP (from 0.78 to 0.79) and a 6% increase in precision (from 0.94 to 1.0), underscoring its enhanced accuracy in small object localization and its ability to identify objects in challenging scenarios. These advancements are attributed to architectural modifications, including the integration of the Squeeze-and-Excitation (SE) module, advanced data augmentation techniques, and multi-scale training, which collectively enhance the model's robustness and effectiveness in addressing the limitations of existing models. These improvements are particularly valuable for applications demanding high accuracy, such as autonomous driving, early-stage cancer

Table 10
Ablation study: Contribution of each modification to SO-YOLOv8's performance.

Model variant	Precision	Recall	mAP@0.5	FLOPs (B) (% Increase)	Params (M)	Inference time (ms/img)
YOLOv8 (Baseline)	0.94	0.89	0.78	257.8	68.2	4.8
Architectural modifications						
+SE Block	0.97	0.89	0.79	261.0 (+1.2%)	69.0	4.6 (−0.2)
Individual data augmentations						
+MixUp	0.96	0.895	0.785	262.0 (+0.4%)	69.2	4.5 (−0.1)
+Copy-Paste	0.98	0.89	0.79	262.3 (+0.6%)	69.3	4.5 (−0.1)
+Random erasing	0.97	0.893	0.79	262.4 (+0.6%)	69.4	4.5 (−0.1)
Full data augmentation						
+All augmentations combined	0.98	0.89	0.79	262.4 (+0.6%)	69.4	4.5 (−0.1)
Optimization techniques						
+Hyperparameter optimization	**1.00**	**0.89**	**0.79**	**263.0 (+0.2%)**	**69.8**	**4.3 (−0.2)**

detection, surveillance, and other real-time monitoring systems. The sensitivity analysis further validates that while hyperparameter tuning contributes to SO-YOLOv8's performance, its improvements remain consistent across a range of reasonable variations. The ablation study confirms that the SE block, data augmentation techniques, and hyperparameter tuning collectively contribute to SO-YOLOv8's performance improvements, with the SE block providing the highest impact on small object detection accuracy. This robustness reveals the significance of the proposed architectural modifications, such as the SE module and multi-scale training, which are central to the model's enhanced performance. SO-YOLOv8 sets a strong foundation for further research, offering a practical and robust solution for real-world scenarios where accurate small object detection is essential.

While SO-YOLOv8 demonstrates meaningful advancements in small object detection, several opportunities remain for further enhancements. Future research will primarily focus on the following principal domains:

- **Optimization of Models for Real-Time and Edge Devices:** The inclusion of SE blocks slightly increases the computational complexity, and training time. Although inference speed remains largely unaffected, real-time deployment on resource-constrained devices may require additional optimizations, such as model pruning or quantization. In order to ensure SO-YOLOv8 remains efficient for edge deployment, we define specific computational constraints. Based on feasibility studies, we aim to limit FLOPs to 50 GFLOPs and parameters to approximately 10 million, ensuring compatibility with low-power devices such as autonomous drones, mobile surveillance systems, and AI accelerators like NVIDIA Jetson Nano and Raspberry Pi. We plan to achieve this through, Model Pruning *(to eliminate redundant parameters while maintaining accuracy, thereby reducing computational burden)*, Quantization *(to reduce the precision of bits, thereby enabling quicker inference on embedded systems with minimal degradation in detection performance)*, and Knowledge Distillation *(Transferring knowledge from SO-YOLOv8 to a smaller, more efficient model)*.
- **Exploring Advanced Attention Mechanisms:** One of the limitation in SO-YOLOv8's performance is in extreme small object scenarios. While it achieves a precision of 1.0, its recall remains unchanged at 0.89, suggesting that some objects are still missed. This trade-off indicates that further research into recall-boosting techniques, such as attention-based refinement layers, may be beneficial. Incorporating advanced attention mechanisms, such as Transformer-based attention, CBAM (Convolutional Block Attention Module), or spatial attention, could refine the model's focus on critical regions, potentially improving mAP while maintaining high precision for small object detection. Also, Transformer-based detection techniques will be considered to improve feature selection, especially for detecting small objects within intricate backgrounds. Further, hybrid CNN-Transformer architectures will be examined, utilizing YOLO's rapid processing alongside the Transformer's enhanced feature extraction skills. These changes seek to sharpen the model's emphasis on critical areas, potentially improving mAP while preserving high precision, so rendering SO-YOLOv8 more resilient in difficult detection contexts.
- **Advancing Hyperparameter Optimization:** While this study employed a grid search methodology for hyperparameter tuning, subsequent research might gain insights from investigating more sophisticated optimization techniques. We plan to explore Bayesian Optimization for adaptive learning rate scheduling, Genetic algorithms for dynamically optimizing architectural hyperparameters, particularly for real-time applications where computational efficiency is crucial, and Neural Architecture Search that possesses considerable potential to enhance model performance while simultaneously decreasing computational expenses. These methodologies may facilitate a more efficient investigation of hyperparameter spaces, especially in relation to intricate duties such as the detection of small objects.
- **Expanding Dataset Generalization:** The PASCAL VOC 2012 dataset was specifically chosen for this study because it contains a relatively low proportion of small objects, making it a challenging testbed for evaluating SO-YOLOv8's ability to enhance small object detection in datasets where small objects are underrepresented. The motivation behind using this dataset was to develop a technique that could improve detection in scenarios where small objects are difficult to identify, rather than selecting a dataset that is already optimized for small object detection. Our results demonstrate that SO-YOLOv8 successfully detects small objects in PASCAL VOC, showcasing its ability to address one of the key limitations of traditional YOLO models. While our primary focus was on evaluating SO-YOLOv8 in this challenging setting, future work will explore its generalization capabilities on more diverse datasets. We have elaborated on specific challenges posed by VisDrone, TinyPerson, and DOTA and how they differ from Pascal VOC. VisDrone is an aerial image dataset where objects vary in size due to altitude differences. It serves as a benchmark for assessing SO-YOLOv8's capability to manage extreme scale variations and detect small objects at different heights. Similarly, TinyPerson is designed for detecting small-scale pedestrians in densely populated urban settings, where occlusion and overlapping objects create significant challenges. Evaluating SO-YOLOv8 on TinyPerson will help determine its effectiveness in identifying objects in crowded environments and addressing occlusion issues, which are crucial for applications such as smart surveillance and autonomous navigation. On the other hand, DOTA (Dataset for Object Detection in Aerial Images) includes satellite imagery containing objects of varying sizes, such as airplanes, ships, and vehicles. Its complex background clutter, extreme scale differences, and diverse object orientations make it an ideal dataset for testing SO-YOLOv8's ability to generalize in aerial detection

tasks. By evaluating the model on these datasets, we can gain deeper insights into its adaptability and identify potential areas for improvement, ensuring better performance in real-world object detection applications.

- **Deployment in Real-World Applications:** Future studies will focus on modifying SO-YOLOv8 for the ongoing tracking of small objects, rendering it appropriate for real-time applications where object continuity is essential. A primary application area is traffic monitoring, wherein the model will be refined to monitor vehicles and people in urban environments, enhancing safety and traffic flow assessment. Moreover, security surveillance is going to benefit from SO-YOLOv8's advancements in real-time tracking, especially in crowd analysis, where the detection and tracking of small objects across several frames is crucial for identifying possible security risks and effectively monitoring public areas. Moreover, video-based detection capabilities will be enhanced by the implementation of temporal consistency techniques to minimize false positives in motion-intensive environments, hence ensuring more consistent detections in dynamic contexts.
- **Ethical and Societal Implications:** SO-YOLOv8 enhances small object detection for applications in safety, healthcare, and automation, it also raises ethical concerns. Improved detection accuracy in surveillance systems may lead to privacy risks if used without proper regulations. There is also a risk of misuse in military applications or unauthorized tracking. To ensure responsible deployment, future work should focus on bias mitigation, privacy-preserving AI techniques, and transparent model interpretability, promoting ethical and fair use of small object detection systems.

CRediT authorship contribution statement

Iqra: Data curation, Conceptualization, Methodology, Software, Investigation, Writing – original draft, Visualization. **Kaisar Javeed Giri:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank the Department of Science & Technology (DST), New Delhi, Govt. of India, for their support under the DST INSPIRE Fellowship Scheme Department of Computer Science, Islamic University of Science and Technology (IUST), for providing the infrastructure and resources necessary for this research. Special thanks to the HPC service at IUST for providing the computational support.

Data availability

I have shared the link of the dataset used in the manuscript.

References

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6154–6162).

Chen, C., Liu, M. Y., Tuzel, O., & Xiao, J. (2017). R-CNN for small object detection. In *Computer vision–ACCV 2016: 13th Asian conference on computer vision, Taipei, Taiwan, November 20-24, 2016, revised selected papers, part v 13* (pp. 214–230). Springer.

Chen, G., Wang, H., Chen, K., Li, Z., Song, Z., Liu, Y., Chen, W., & Knoll, A. (2020). A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(2), 936–953.

Chen, C., Zhang, Y., Lv, Q., Wei, S., Wang, X., Sun, X., & Dong, J. (2019). Rrnet: A hybrid detector for object detection in drone-captured images. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*.

Cheng, M., Ma, H., Ma, Q., Sun, X., Li, W., Zhang, Z., Sheng, X., Zhao, S., Li, J., & Zhang, L. (2023). Hybrid transformer and cnn attention network for stereo image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1702–1711).

Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, *82*(6), 9243–9275.

Du, J. (2018). Understanding of object detection based on CNN family and YOLO. In *Journal of physics: conference series, vol. 1004*. IOP Publishing, Article 012029.

Farooq, J., Muaz, M., Khan Jadoon, K., Aafaq, N., & Khan, M. K. A. (2024). An improved YOLOv8 for foreign object debris detection with optimized architecture for small objects. *Multimedia Tools and Applications*, *83*(21), 60921–60947.

Girshick, R. (2015). Fast r-cnn. arxiv 2015. arXiv preprint arXiv:1504.08083.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

He, X., Cheng, R., Zheng, Z., & Wang, Z. (2021). Small object detection in traffic scenes based on YOLO-MXANet. *Sensors*, *21*(21), http://dx.doi.org/10.3390/s21217422, URL: https://www.mdpi.com/1424-8220/21/21/7422.

He, Y., Su, Y., Wang, X., Yu, J., & Luo, Y. (2023). An improved method MSS-YOLOv5 for object detection with balancing speed-accuracy. *Frontiers in Physics*, *10*, Article 1101923.

Hemmati, A., & Rahmani, A. M. (2024). ClusFC-IoT: A clustering-based approach for data reduction in fog-cloud-enabled IoT. *Concurrency and Computation: Practice and Experience*, *36*(27), Article e8284.

Hemmati, A., & Zarei, M. (2024). UFC3: UAV-aided fog computing based congestion control strategy for emergency message dissemination in 5G internet of vehicles. *Automotive Innovation*, *7*(3), 456–472.

Henderson, P., & Ferrari, V. (2017). End-to-end training of object class detectors for mean average precision. In *Computer vision–ACCV 2016: 13th Asian conference on computer vision, Taipei, Taiwan, November 20-24, 2016, revised selected papers, part v 13* (pp. 198–213). Springer.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, J., & Li, T. (2024). Small object detection by DETR via information augmentation and adaptive feature fusion. In *Proceedings of 2024 ACM ICMR workshop on multimodal video retrieval* (pp. 39–44).

Ji, C. L., Yu, T., Gao, P., Wang, F., & Yuan, R. Y. (2024). Yolo-tla: An efficient and lightweight small object detection model based on YOLOv5. *Journal of Real-Time Image Processing*, *21*(4), 141.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8. URL: https://github.com/ultralytics/ultralytics, AGPL-3.0 License.

Ju, B., Yang, W., Jia, J., Ye, X., Chen, Q., Tan, X., Sun, H., Shi, Y., & Ding, E. (2021). Danet: Dimension apart network for radar object detection. In *Proceedings of the 2021 international conference on multimedia retrieval* (pp. 533–539).

Kaur, P., Khehra, B. S., & Mavi, E. B. S. (2021). Data augmentation for object detection: A review. In *2021 IEEE international midwest symposium on circuits and systems* (pp. 537–543). IEEE.

Lavie, A., Sagae, K., & Jayaraman, S. (2004). The significance of recall in automatic metrics for MT evaluation. In *Machine translation: from real users to research: 6th conference of the association for machine translation in the Americas, AMTA 2004, Washington, DC, USA, September 28-October 2, 2004. proceedings 6* (pp. 134–143). Springer.

Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision* (pp. 734–750).

Li, Y., Chen, Y., Wang, N., & Zhang, Z. (2019). Scale-aware trident networks for object detection. arXiv preprint arXiv:1903.08589.

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2017). Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. arXiv preprint arXiv:1708.02002.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, the Netherlands, October 11–14, 2016, proceedings, part i 14* (pp. 21–37). Springer.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Liu, M., Wang, X., Zhou, A., Fu, X., Ma, Y., & Piao, C. (2020). UAV-YOLO: Small object detection on unmanned aerial vehicle perspective. *Sensors*, *20*(8), http://dx.doi.org/10.3390/s20082238, URL: https://www.mdpi.com/1424-8220/20/8/2238.

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). DC-YOLOv8: Small-size object detection algorithm based on camera sensor. *Electronics*, *12*(10), 2323.

Lu, Y., Chen, X., Wu, Z., Tan, M., & Yu, J. (2023). Binary similarity few-shot object detection with modeling of hard negative samples. *IEEE Transactions on Multimedia*.

Mahaur, B., Mishra, K., & Kumar, A. (2023). An improved lightweight small object detection framework applied to real-time autonomous driving. *Expert Systems with Applications*, *234*, Article 121036.

Niu, Y., Cheng, W., Shi, C., & Fan, S. (2023). YOLOv8-CGRNet: A lightweight object detection network leveraging context guidance and deep residual learning. *Electronics*, *13*(1), 43.

Özcan, İ., Altun, Y., & Parlak, C. (2024). Improving YOLO detection performance of autonomous vehicles in adverse weather conditions using metaheuristic algorithms. *Applied Sciences*, *14*(13), 5841.

Pan, J., Xu, S., Cheng, Z., & Lian, S. (2024). C2F-YOLO: A coarse-to-fine object detection framework based on YOLO. In *Proceedings of the 2024 3rd Asia conference on algorithms, computing and machine learning* (pp. 150–157).

Pham, M. T., Courtrai, L., Friguet, C., Lefèvre, S., & Baussard, A. (2020). YOLO-fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*, *12*(15), http://dx.doi.org/10.3390/rs12152501, URL: https://www.mdpi.com/2072-4292/12/15/2501.

Rahimi, M., Mostafavi, M., & Arabameri, A. (2024). Automatic detection of brain tumor on MRI images using a YOLO-based algorithm. In *2024 13th Iranian/3rd international machine vision and image processing conference* (pp. 1–5). IEEE.

Ranawana, R., & Palade, V. (2006). Optimized precision-a new measure for classifier performance evaluation. In *2006 IEEE international conference on evolutionary computation* (pp. 2254–2261). IEEE.

Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149.

Setiyono, B., Amini, D. A., & Sulistyaningrum, D. R. (2021). Number plate recognition on vehicle using YOLO-Darknet. In *Journal of physics: conference series, vol. 1821*. IOP Publishing, Article 012049.

Shetty, S. (2016). Application of convolutional neural network for image classification on pascal VOC challenge 2012 dataset. arXiv preprint arXiv:1607.03785.

Suhartono, S., Zain, S. G., & Ardilla, A. (2024). Detection of vehicle type and license plate with convolutional neural network model YOLOv7. *Jurnal Teknik Informatika (JUTIF)*, *5*(2), 621–636.

Sun, W., Dai, L., Zhang, X., Chang, P., & He, X. (2022). RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 1–16.

Tian, D., Zhao, J., Zhang, Z., Yu, H., Sun, M., & Liu, H. (2022). Absolute size IoU loss for the bounding box regression of the object detection. *Neurocomputing*, *500*, 1029–1040.

Tong, K., Wu, Y., & Zhou, F. (2020). Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, *97*, Article 103910.

Torralba, A., Fergus, R., & Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(11), 1958–1970.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, *2018*(1), Article 7068349.

Wang, Z. Z., Xie, K., Zhang, X. Y., Chen, H. Q., Wen, C., & He, J. B. (2021). Small-object detection based on YOLO and dense block via image super-resolution. *IEEE Access*, *9*, 56416–56429. http://dx.doi.org/10.1109/ACCESS.2021.3072211.

Xianbao, C., Guihua, Q., Yu, J., & Zhaomin, Z. (2021). An improved small object detection method based on Yolo V3. *Pattern Analysis and Applications*, *24*, 1347–1355.

Xiao, Y., Lepetit, V., & Marlet, R. (2022). Few-shot object detection and viewpoint estimation for objects in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 3090–3106.

Yang, F., Fan, H., Chu, P., Blasch, E., & Ling, H. (2019). Clustered object detection in aerial images. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8311–8320).

Yang, J., Wu, C., Du, B., & Zhang, L. (2021). Enhanced multiscale feature fusion network for HSI classification. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(12), 10328–10347.

Yu, X., Gong, Y., Jiang, N., Ye, Q., & Han, Z. (2020). Scale match for tiny person detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1257–1265).

Zhang, J., Xia, K., Huang, Z., Wang, S., & Akindele, R. G. (2023). ETAM: Ensemble transformer with attention modules for detection of small objects. *Expert Systems with Applications*, *224*, Article 119997.

Zhang, Y., Zhang, H., Huang, Q., Han, Y., & Zhao, M. (2024). DsP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects. *Expert Systems with Applications*, *241*, Article 122669.

Zhao, L., Zhi, L., Zhao, C., & Zheng, W. (2022). Fire-YOLO: a small target object detection method for fire inspection. *Sustainability*, *14*(9), 4930.

Zhao, H., Zhou, Y., Zhang, L., Peng, Y., Hu, X., Peng, H., & Cai, X. (2020). Mixed YOLOv3-LITE: A lightweight real-time object detection method. *Sensors*, *20*(7), 1861.

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2110–2118).

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, *111*(3), 257–276.