

CAM-YOLO: A Framework for Non-Salient Object Detection by Exploiting Contextual Interdependencies

Chengang Dong^{1,2}, Yongkang Ding² and Jianwei Hu^{1,3,*}

¹ School of Mathematics and Statistics, Huangshan University, Huangshan, China

² College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

³ Huangshan Technology Innovation Center for Digital Economy and Big Data Analysis, Huangshan, China

INFORMATION

Keywords:

Object detection
feature enhancement
contextual modeling
YOLOv8

DOI: 10.23967/j.rimni.2026.10.76340

CAM-YOLO: A Framework for Non-Salient Object Detection by Exploiting Contextual Interdependencies

Chengang Dong^{1,2}, Yongkang Ding² and Jianwei Hu^{1,3,*}

¹School of Mathematics and Statistics, Huangshan University, Huangshan, China

²College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

³Huangshan Technology Innovation Center for Digital Economy and Big Data Analysis, Huangshan, China

ABSTRACT

Real-time object detection requires identifying objects in video streams or consecutive images with minimal latency, yet it continues to struggle with non-salient objects—those that are small, occluded, or otherwise inconspicuous. To address this limitation, this paper proposes CAM-YOLO, an enhanced architecture based on YOLOv8. First, to mitigate the baseline model's limited representational capacity for non-salient targets, we introduce a Multi-Scale Aggregation Module (MSAM) into the feature fusion process, enabling the backbone network to extract more discriminative fine-grained features. Second, to better capture global contextual relationships associated with such objects, we design a Contextual Association Module (CAM) that explicitly models long-range spatial dependencies. Furthermore, we integrate a Dual-Branch Attention Mechanism (DBAM) to refine the feature processing flow, thereby strengthening the contextual feature representations crucial for detecting non-salient instances. Extensive experiments on two large-scale public benchmarks, Microsoft Common Objects in Context 2017 (MS COCO 2017) and PASCAL Visual Object Classes (PASCAL VOC), demonstrate that CAM-YOLO achieves highly competitive performance compared to several widely-adopted real-time detectors.

OPEN ACCESS

Received: 19/11/2025

Accepted: 15/01/2026

Published: 21/07/2026

DOI

10.23967/j.rimni.2026.10.76340

Keywords:

Object detection
feature enhancement
contextual modeling
YOLOv8

1 Introduction

Considering the current trajectory of artificial intelligence development, machine vision has emerged as one of the key domains in information technology, where real-time object detection, as a core task, has garnered widespread attention. The objective of real-time object detection is to efficiently categorize and localize target objects in video streams or continuous image sequences, finding extensive applications in video surveillance, autonomous driving, unmanned aerial vehicle monitoring, human-computer interaction, and augmented reality. In recent years, deep learning architectures, specifically Convolutional Neural Networks (CNNs) [1], have driven remarkable progress in this field. However, detecting inconspicuous objects remains challenging, as their perceptually subtle differences from the surroundings, small size, or similarity in spectral properties to the environment often leads to misidentification, thereby compromising detection accuracy. Although traditional YOLO series

methods [2–5] enable rapid inference, they frequently suffer from missed detections for inconspicuous objects, as illustrated in Fig. 1. The targets marked by yellow boxes in the aforementioned two figures are not precisely determined owing to their diminutive dimensions or obstruction, which restricts their feature representation.

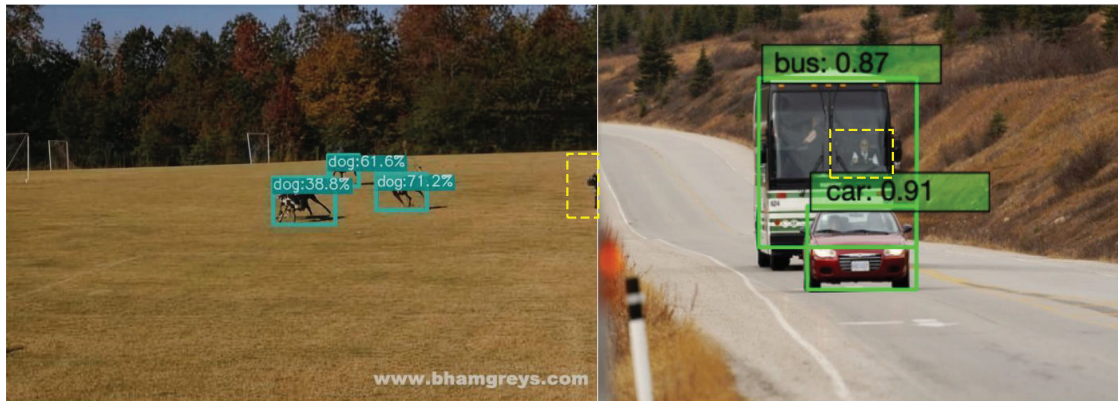


Figure 1: Missed detection issues throughout conventional detection pipelines

YOLOv8 [6], marks a considerable leap in capabilities for the YOLO lineage, incorporates multiple innovative technologies to enhance model performance. By incorporating the C2f module for backbone refinement and adopting an anchor-free paradigm [2] coupled with a task-aligned assignment strategy [7] to streamline the architecture of the detection head, it achieves better alignment between classification and localization tasks, significantly improving detection accuracy and bounding box regression stability. However, due to limitations in feature extraction and contextual semantic modeling, YOLOv8 still exhibits considerable room for improvement in detecting inconspicuous objects.

In order to tackle these issues, this study presents an innovative object detection framework named CAM-YOLO. Building upon YOLOv8 [6] as the baseline, CAM-YOLO further enhances the detection capability for inconspicuous objects through two key architectural improvements. In the experimental section, we evaluate CAM-YOLO against cutting-edge real-time object detection techniques using two publicly available datasets, MS COCO [8] and PASCAL VOC [9], to verify its superior accuracy. The principal advancements presented of this study are delineated below:

- Firstly, to enable the model to acquire richer feature representations, we refine the feature fusion strategy in YOLOv8 via the implementation of Multi-Scale Aggregation Module (MSAM), which facilitates the extraction of more fine-grained features pertinent to inconspicuous targets.
- Secondly, in order to further enhance the contextual textual feature representation associated with inconspicuous targets, we improve the backbone architecture of YOLOv8 by incorporating a Contextual Association Module (CAM), thereby enabling the model to model long-range spatial interactions among features from different regions of the image.
- Finally, we present an innovative visual attention mechanism termed the Dual-Branch Attention Mechanism (DBAM). By modeling multi-scale attention in both spatial and channel branches for the received features, DBAM further enhances detection performance.

2 Literature Review

2.1 Real-Time Object Detection Models

Real-time object detection models utilize the features of the complete image as inputs and generates the final predictions of object categories and bounding boxes, a process illustrated in Fig. 2. These approaches leverage CNN or Transformer models [10,11] to learn hierarchical image representations, and employ fully connected networks for classification and regression tasks. YOLOv5 [12], Representing a canonical iteration in the YOLO series, this architecture employs a deeper DarkNet-53 [13] backbone to improve detection reliability. However, it may still encounter limitations in precision when detecting small objects within complex backgrounds. YOLOv7 [12] harnesses an Efficient Layer Aggregation Network (ELAN) in order to facilitate the aggregation of features across multiple scales and incorporates the structural re-parameterization concept from RepVGG [14] to boost inference speed. Nevertheless, YOLOv7's anchor-based sample matching methodology demands additional optimization to enhance proposal box quality and localization accuracy for inconspicuous objects. YOLOv8 [6] better integrates the advantages of previous YOLO models through the utilization of a decoupled head in conjunction with an anchor-free mechanism, thereby subsequently advancing both the performance and computational efficiency of the detector. Nonetheless, when the feature representation of a object is limited, the model often struggles to achieve accurate regression.

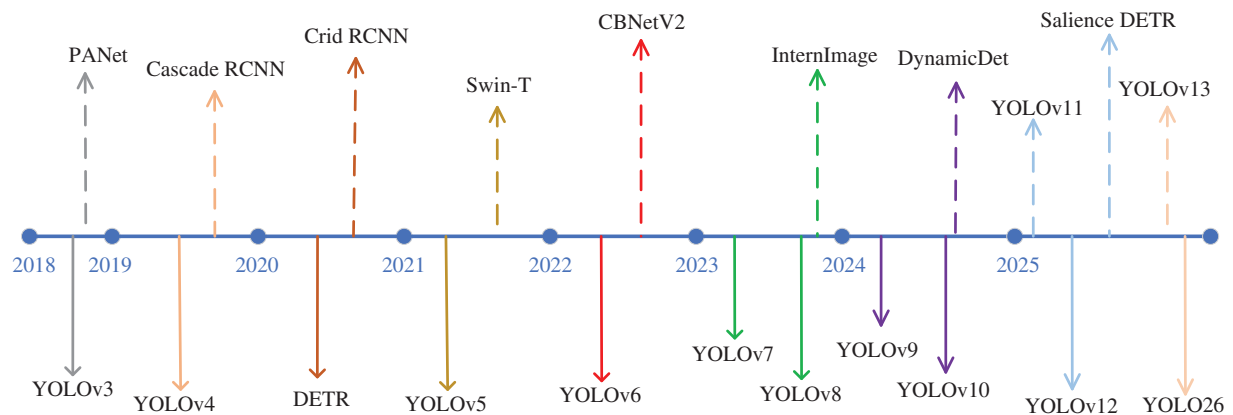


Figure 2: Recent advances in neural network-based object detection models

YOLOv10 [15] realizes efficient real-time object detection through an anchor-free design and hierarchical feature fusion, realizing considerable a dual-faceted optimization of accuracy and runtime efficiency in comparison to previous versions and supporting end-to-end deployment. However, the integration of more sophisticated architectural components and refined feature integration schemes also results in increased model complexity. Aiming for a model that is both accurate and fast in real-world deployment, YOLOv12's [16] designers engineered key architectural refinements. This focus translates to robust performance on standard benchmarks like COCO, achieved through techniques such as enhanced CSP modules and structural re-parameterization. For practitioners, this translates to welcome computational efficiency and easier deployment. Yet one could argue that these improvements are incremental; consequently, for specialized tasks like microscopic cell detection, the performance leap over earlier versions might be less pronounced. What about detecting objects in heavily occluded or crowded scenes? Here, the approach shifts: YOLOv13 [17] integrates attention mechanisms or lightweight Transformer blocks to refine feature representation for such complexity. While this boosts its potential robustness against occlusions—a common headache in

autonomous driving perception—this comes at a cost. The model’s complexity and computational overhead rise significantly, often demanding high-end GPUs and more meticulous tuning to reach optimal performance.

Furthermore, certain methods based on the transformer architecture have also gained widespread recognition. RetinaNet [18] constitutes a methodological innovation in single-stage detection by addressing class imbalance via a focal loss mechanism, thereby significantly enhancing recognition performance on small and challenging instances. Nevertheless, the generalizability of RetinaNet can be constrained by its preset anchor designs, demanding non-trivial manual adjustment to maintain detection accuracy across varying object scales and densities. Detection Transformer (DETR) [19] introduces a paradigm-shifting framework that reformulates object detection as an end-to-end set prediction problem via Transformer encoders. However, its performance tends to degrade under sparse or dense occlusion scenarios, highlighting limitations in existing set-based prediction paradigms that warrant further investigation. RT-DETR [20] establishes a unified architecture that integrates object detection and tracking within a singular Transformer-based framework for joint feature representation and association learning. Nevertheless, the propensity of Transformer architectures toward overfitting persists as an unresolved challenge in this unified paradigm.

2.2 Visual Attention Mechanisms

In deep learning-based computer vision tasks, visual attention mechanisms serve as effective techniques to enhance neural networks’ focus on salient information within images. The Convolutional Block Attention Module (CBAM) [21] adaptively modulates the importance of feature maps via dual-axis feature modulation mechanisms (channel-wise and spatial), thereby improving the framework’s representational capacity and performance, albeit with corresponding compromises in operational efficiency and implementation intricacy. The Global Attention Mechanism (GAM) [22] exploits overarching contextual information to dynamically recalibrate input feature weights, highlighting salient features and suppressing irrelevant information. However, as GAM typically computes attention weights via linear or non-normalized approaches, it may lead to skewed attention distributions. The Coordinate Attention Mechanism [23] incorporates both the channel-wise and spatial components of the feature matrix, learning weights for every pixel to empower the network to dynamically concentrate on essential features. Nevertheless, since this mechanism lacks explicit semantic modeling capabilities, it may fail to sufficiently capture semantic-related features required for instance segmentation tasks.

3 Methodology

3.1 Structure Design

This study introduces a target detection model referred to as CAM-YOLO, which optimizes the detection performance of less prominent targets in images by fine-tuning the feature processing approach of the YOLOv8x network. The integrated framework of CAM-YOLO is depicted in Fig. 3, where the end-to-end structural design simplifies the training process.

CAM-YOLO achieves the following three optimizations based on the YOLOv8 network: **(a)** CAM-YOLO introduces a Multi-Scale Attention Module (MSAM) at the backbone’s terminal stage to enhance model capacity for capturing more detailed features from shallow layers. **(b)** CAM-YOLO incorporates a Context-Aware Module (CAM) in the neck network to better model the contextual relationships between different spatial domains. **(c)** CAM-YOLO optimizes the model’s information processing by introducing a Dual-Branch Attention Module (DBAM), further enhancing the model’s feature abstraction.

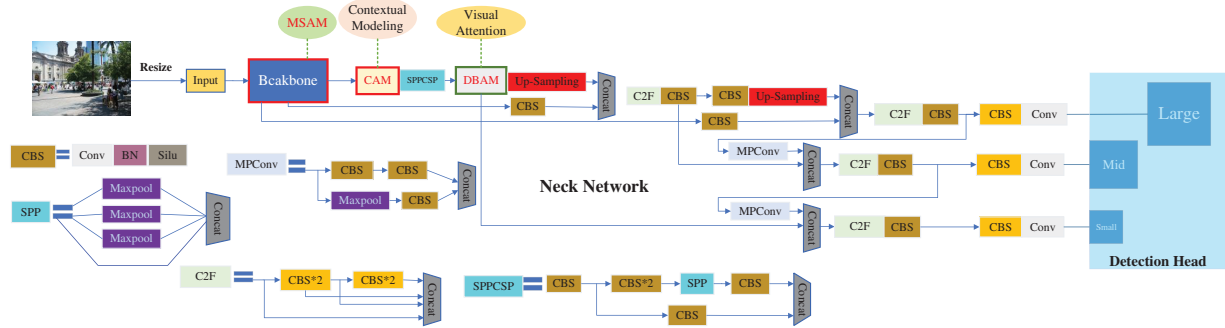


Figure 3: Comprehensive architectural framework of CAM-YOLO

3.2 Multi-Scale Aggregation Module

To optimize the model's discriminative capability for small objects within images, the current section introduces the Multi-Scale Attention Module (MSAM). Typically, shallow feature maps contain shallow features retain finer textural details, whereas deep features encapsulate broader structural patterns and semantic concepts. Although deep feature maps have a larger receptive field, they often exhibit lower resolution and tend to lack detailed information, which adversely affects the detection performance of less prominent targets. To address this concern, the current section includes the MSAM module into the YOLOv8 baseline, replacing the original Backbone's neck network's final CSP Bottleneck with 2 Convolutions (C2f) structure, thus aggregating more detailed features from the shallow network.

Specifically, the hierarchical structure of the model's backbone network is roughly defined as $\{P_1, \dots, P_5\}$. MSAM conceptually inspired by the Atrous Spatial Pyramid Pooling (ASPP) [24] approach by utilizing dilated convolutions to amalgamate hierarchical representations encompassing both detailed and semantic information. Its approximate internal structure is depicted in Fig. 4. The information processing framework of the MSAM module can be mathematically formulated as Eq. (1).

$$R = \lambda_1 \text{Pooling}[DConv_{7 \times 7}(P_1)] + \lambda_2 DConv_{5 \times 5}(P_2) + \lambda_3 Conv_{3 \times 3}(P_3) + \lambda_4 Conv_{1 \times 1}(P_4) \quad (1)$$

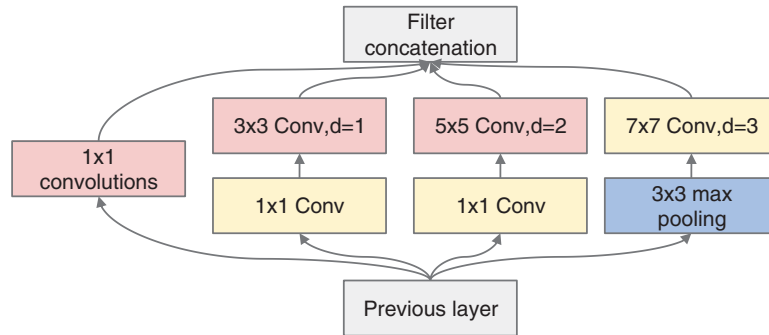


Figure 4: The feature processing method within the MSAM module

The coefficients λ_i serve to balance contributions across different branches. To achieve this, a compact channel attention mechanism dynamically generates their values, allowing λ_i to adapt based on each image's global content. Consider a scene containing numerous non-salient objects

demanding fine detail—here, the mechanism automatically favors branches with smaller receptive fields. Conversely, when broader contextual information predominates, it shifts weighting toward branches with larger receptive fields. We note this adaptive approach significantly enhances robustness, particularly improving performance in complex, cluttered environments where non-salient objects might otherwise be missed. Equally important, the design introduces negligible parameter overhead and maintains inference efficiency, fully satisfying real-time detection requirements.

The term *Pooling*[·] denotes a 3×3 max pooling operation, *Conv*(·) indicates a convolution operation, and *DConv*(·) signifies a dilated convolution operation. The MSAM module employs multiple branches with varying dilation rates to fuse feature layers of different scales and depths, thereby promoting information exchange among various receptive fields. The multi-branch feature fusion design enables the retention of more shallow texture features, assisting CAM-YOLO in better handling less prominent targets within images.

3.3 Contextual Association Module

The YOLOv8 baseline model typically heavily utilizes successive convolutional stages for progressive feature learning. However, this stacked convolutional architecture often struggles to efficiently capture global contextual features in the spatial domain due to its limited local receptive fields and inadequate context modeling over extended ranges, thereby restricting the model's feature representation capability. To effectively capture long-range contextual relationships in the spatial domain, CAM-YOLO introduces the Contextual Association Module (CAM), whose general implementation approach is illustrated in Fig. 5.

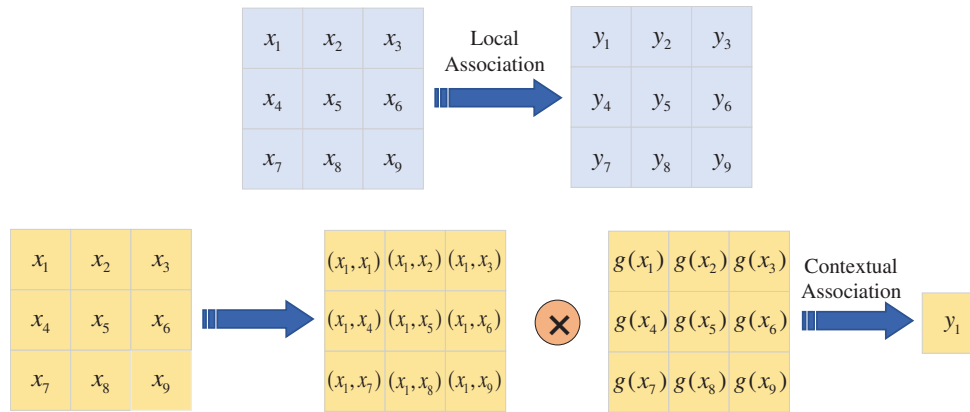


Figure 5: Illustration of spatial domain contextual correlation modeling

To quantitatively determine the information for each region of the output feature matrix, CAM must consider every region in the input feature matrix, thereby assessing their intrinsic relationships in a manner akin to the modeling process of self-attention mechanisms. CAM further excavates the potential semantic information within feature maps through the identification of extended relationships among diverse spatial areas within the image [25]. Specifically, the CAM module integrates feature map serialization with a self-attention mechanism, enabling efficient modeling of long-range spatial dependencies within the convolutional architecture. The mathematical formulation of this approach is provided in Eq. (2).

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \quad (2)$$

In the above equation, x denotes the vectorized form of the input feature matrix via a flattening operation. Indices (i, j) denote the index of the current feature point within the dimensional layout of this vector, while y reflects the globally mapped output at the target location. The term $C(\cdot)$ denotes a normalization operation, which in this study is implemented using the softmax function. The function $f(\cdot)$ serves to compute the inter-regional affinity of features across distinct locations, and a Gaussian function is employed for this purpose, as shown in Eq. (3).

$$f(x_i, x_j) = e^{\theta(x_i)^T \rho(x_j)} \quad (3)$$

Additionally, in Eq. (2), $g(\cdot)$ implements a projection operation on the features within the current region, which is realized through a spatial 1×1 convolution operation W_g . This mapping process can be formulated as Eq. (4).

$$g(x_j) = W_g x_j \quad (4)$$

The complete implementation of CAM is mathematically summarized in Eq. (5).

$$y_i = \text{soft max}(\theta(x_i)^T \rho(x_j)) g(x_j) = \frac{1}{\sum e^{\theta(x_i)^T \rho(x_j)}} e^{\theta(x_i)^T \rho(x_j)} W_g x_j \quad (5)$$

CAM establishes associations between remote pixels in the image, empowering the model's capacity to fully leverage global contextual feature, thus improving the detection precision of the YOLOv8 baseline model. The specific internal information processing processes are depicted in Fig. 6. To model long-range spatial dependencies, CAM distinguishes itself from standard Transformer self-attention [26] in three key aspects. First, it employs a Gaussian kernel for similarity measurement—instead of dot-product—to better align with the structured nature of CNN feature maps, preserving spatial continuity. Second, CAM requires no explicit positional encoding; the translational equivariance inherent in convolutional operations provides implicit positional awareness. Third, architecturally, CAM integrates as a lightweight plugin within the neck network, specifically designed to augment the CNN backbone's capacity for global context modeling. Compared to classical non-local blocks [27], CAM offers enhanced stability in feature association through its consistent use of a Gaussian kernel. Furthermore, by deeply integrating with the multi-scale feature pyramid—particularly during cross-scale interactions—it strengthens feature representation for non-salient targets while maintaining computational efficiency.

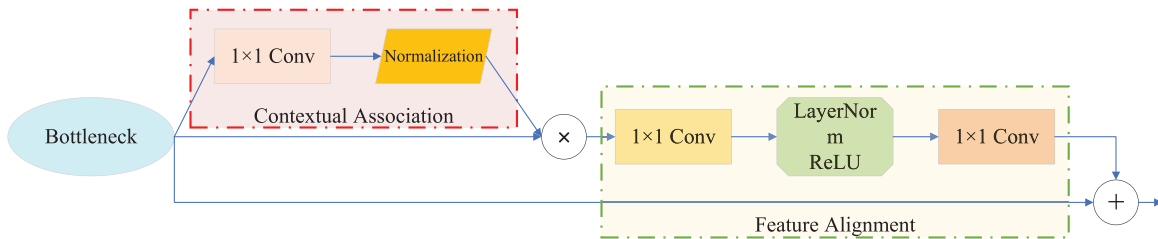


Figure 6: The specific implementation method of the contextual association module

3.4 Dual-Branch Attention Mechanism

The single-channel or spatial modeling visual attention mechanisms typically lack the capability to adequately characterize the sophisticated dependencies and contextual information among features, restricting the framework's effectiveness in managing various scenarios. To further optimize the processing of contextual information related to less prominent targets in YOLOv8, this section introduces a visual attention module named the Dual-Branch Attention Module (DBAM).

Specifically, DBAM divides the attention modeling process into two concurrent pathways: the spatial modeling branch and the channel modeling branch, as illustrated in Fig. 7. In the channel modeling process, this branch receives the feature map $Z(H \times W \times C)$ input from the Bottleneck, which is first processed through Global Average Pooling across the channel axis to derive the initial channel response signal. Subsequently, we employ a Channel Shuffle operation to achieve a more enriched cross-channel feature representation, resulting in a set of initial channel descriptors. Following this, a functional composition of activation and linear transformation are used to train the appropriate channel descriptor vector, which is finally aligned in dimension through a 1×1 convolution and batch normalization. The channel modeling process is digitally represented by Eq. (6).

$$D = Fus\left(\forall_{i \in (1 \dots C)} \left\{ \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_i(i, j) \right\}\right) \quad (6)$$

In this equation, D denotes the channel descriptors obtained from the preliminary processing, and $Fus(\cdot)$ signifies the Channel Shuffle operation, while $x_i(i, j)$ represents the pixel data items. Subsequently, we adjust the channel descriptors according to Eq. (7).

$$Att_c = Batch_Norm(Conv_{1 \times 1}(FC(ReLU(D)))) \quad (7)$$

where Att_c represents the channel attention modeling, $Conv_{1 \times 1}$ denotes the 1×1 convolution operation, $FC(\cdot)$ indicates the fully connected mapping, $ReLU(\cdot)$ represents the nonlinear activation function, and batch-norm refers to the batch normalization operation.

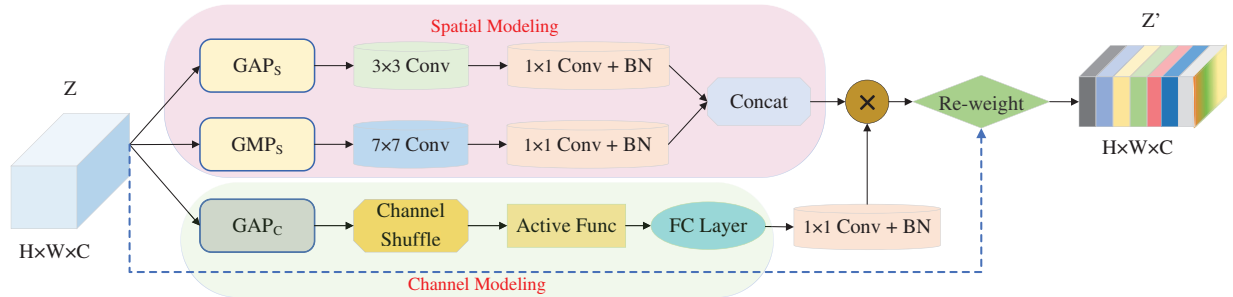


Figure 7: Diagram illustrating the dual-branch attention modeling of DBAM

In the spatial modeling branch, DBAM captures contextual information across different receptive fields by employing two different global pooling strategies combined with two types of convolution operations, as specified in Eqs. (8) and (9).

$$Att_{s_1} = Batch_Norm(Conv_{1 \times 1}(Conv_{3 \times 3}(GAP_s))) \quad (8)$$

$$Att_{s_2} = Batch_Norm(Conv_{1 \times 1}(Conv_{7 \times 7}(GMP_s))) \quad (9)$$

Here, Att_{s_1} and Att_{s_2} denote two specifications of spatial attention modeling, while $Conv_{3 \times 3}$ and $Conv_{7 \times 7}$ represent 3×3 and 7×7 convolution processing, respectively, and GAP_s and GMP_s indicate two different global pooling strategies. GAP_c is the operation that performs global average pooling along the channel dimension of a feature map. Subsequently, we utilize a Concat operation to merge spatial signals from different receptive fields. This relationship can be mathematically expressed as shown in Eq. (10).

$$Att_s = Concat[Att_{s_1} \oplus Att_{s_2}] \quad (10)$$

Finally, DBAM combines the attention modeling results from both branches to obtain the final output feature Z' . By dynamically fusing multidimensional features—semantic significance across decoupled dimensions: channel-wise characteristics and spatial locality. Thus, DBAM achieves a more refined feature selection and cross-dimensional interaction. Algorithm 1 presents the specific steps for implementing DBAM in the form of pseudocode.

Algorithm 1 : PyTorch-like code for dual-branch attention mechanism

```

1: Def Attn_DBAM(Z) # Global average pooling
2: # Channel modeling branch
3: avg_pooled = global_avg_pool(Z) # Global average pooling
4: shuffled = channel_shuffle(avg_pooled) # Channel shuffling
5: activated_vector = activated(shuffled) # Activation function applied
6: channel_vector = fc_layer(activated_vector) # Fully connected layer to generate channel
   description
7: channel_att = Conv1×1_bn(channel_vector) # 1×1 convolution with batch normalization
8: # Spatial modeling branch
9: spatial_avg = global_avg_pool(Z) # Global average pooling
10: spatial_max = global_max_pool(Z) # Global max pooling
11: spatial_att1 = Conv1×1_bn(Conv3×3(spatial_avg)) # Average pooling + 3×3 convolution
12: spatial_att2 = Conv1×1_bn(Conv7×7(spatial_max)) # Max pooling + 7×7 convolution
13: spatial_att = concat(spatial_att1, spatial_att2) # Concatenation of feature maps
14: final_att = channel_att * spatial_att # Fuse attention from both branches
15: output = Z * final_att # Apply attention weighting to the original feature map
16: Return output(Z') # Return the weighted feature map

```

4 Experimental Validation

This chapter designed several comparative analyses to assess the efficacy of CAM-YOLO across multiple metrics on the MS COCO 2017 and PASCAL VOC public datasets. Additionally, to validate the effects of the three proposed improvement modules (MSAM, CAM, and DBAM), this chapter also designed several ablation experiments to investigate their individual effectiveness.

4.1 Experimental Setup

The implementation environment for training CAM-YOLO is based on an NVIDIA RTX 4090 graphics card, operating on Ubuntu 18.04 LTS, and implemented using the PyTorch deep learning architecture for model training and optimization. Furthermore, NVIDIA CUDA technology is utilized for GPU acceleration to enhance computational efficiency. We briefly describe the benchmark datasets used in our study as follows:

- MS COCO 2017 [8] dataset is a prominent resource in the field of computer vision, comprising more than 118,000 images designated for training purposes and an additional 5000 images intended for validation, encompassing a total of 80 distinct object categories. It includes bounding boxes, instance segmentation masks, and diverse annotations, enabling activities such as object localization and the generation of descriptive captions for images. The dataset's complexity, with multiple objects and varying scales, enhances algorithmic challenges and generalization. As a benchmark for evaluating visual algorithms, MS COCO 2017 drives advancements in deep learning technologies.
- PASCAL VOC [9] is a foundational dataset in computer vision, covering 20 object categories with high-quality annotations for classification, detection, and segmentation. The most utilized versions are those from 2007 and 2012, featuring approximately 10,000 annotated real images. PASCAL VOC provides a unified benchmark for algorithm evaluation, significantly advancing object recognition and segmentation, and is essential for validating deep learning methods.

4.1.1 Performance Metrics

When assessing the performance of object detection framework, the commonly used metric is mean Average Precision (mAP), calculated as the average across ten thresholds of Intersection over Union (IoU) ranging from 0.50 to 0.95. To more comprehensively evaluate the model's improvement on missed detections, we adopt the Average Recall (AR) metric to holistically assess the model's recall capability, thereby verifying whether more inconspicuous targets can be detected by the model. Generally, less prominent objects in images occupy a smaller pixel area. To evaluate the framework's perceptual capacity to identify objects of different dimensions, the MS COCO dataset introduces sub-scale metrics, such as AP^S (mean precision for minor objects with pixel areas less than 32^2) and AP^M (mean precision for intermediate-sized objects with pixel areas between 32^2 and 96^2), to systematically assess the efficacy of the model in identifying objects of small and intermediate sizes. Additionally, real-time capacity is commonly evaluated with frames per second (FPS), which holds significant reference value for practical application scenarios.

4.1.2 Training

The CAM-YOLO framework is developed extending the YOLOv8x implementation [6] and utilizes the Mosaic [28] and CutMix [29] sample expansion methods to boost generalization capability. Throughout the data conditioning phase, source imagery are consistently scaled to 640×640 pixels, through dimensional standardization operations. The model utilizes the recently released optimizer Lion [30] for training, with the upper limit training duration set to 240 epochs. Although SGD and AdamW are more common in object detection, we observed that Lion offers superior convergence efficiency and stability during training—particularly in the early phases—as illustrated in Fig. 8. Theoretically, Lion combines a sign function with momentum; this design enhances robustness to gradient noise and facilitates efficient escape from saddle points in the loss landscape. Consequently, it steers optimization toward a stable solution more rapidly. We therefore based our choice primarily on Lion's marked advantage in accelerating training convergence, rather than on an expectation that it would decisively improve final detection accuracy.

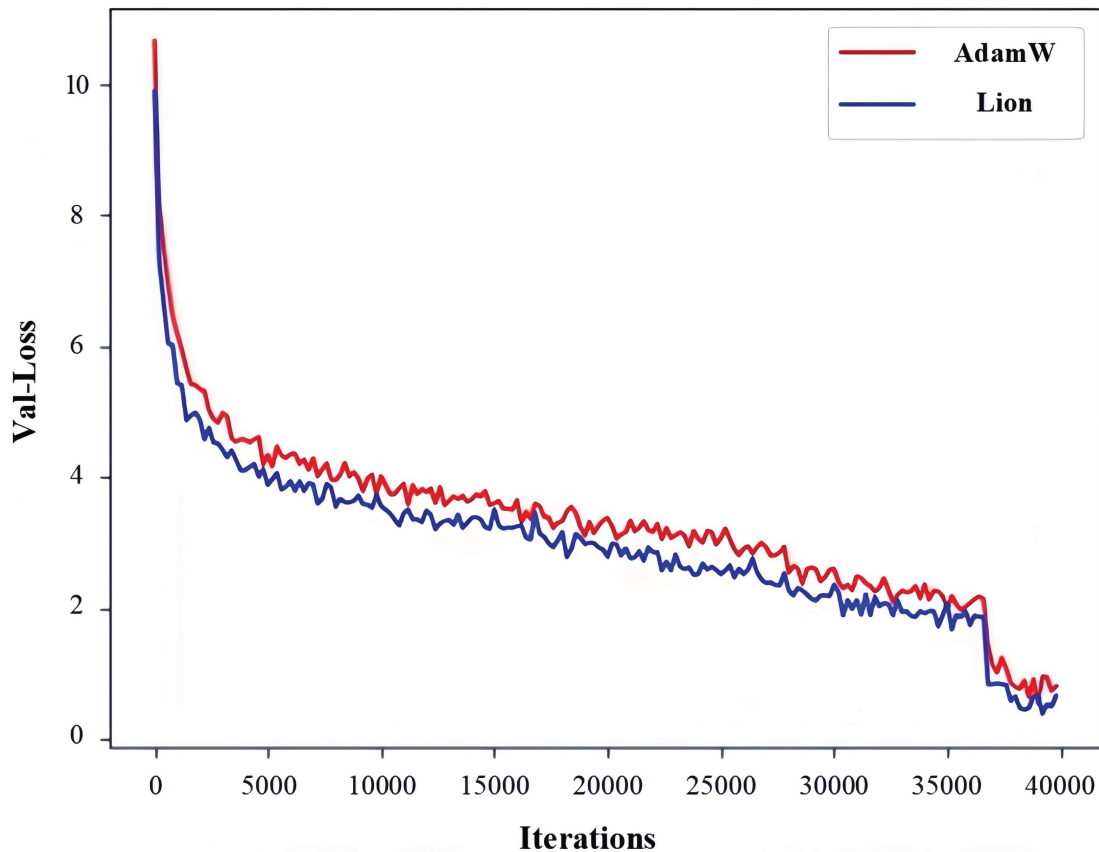


Figure 8: Comparative analysis of convergence behavior: Lion vs. AdamW optimizers

The beginning parameter update rate is established at $5e-3$, adjusted to $5e-4$ at the 180th epoch, and subsequently decreased to $5e-5$ at the 210th epoch. Furthermore, to maintain the convergence robustness and efficacy of the learning process, it is imperative that the lower limit batch size is established at 20. Fig. 9 depicts the dynamics of classification loss (cls_loss), localization loss (box_loss), average recall (AR), and AP over the learning process of CAM-YOLO framework.

4.2 Findings from Experiments Using Public Datasets

Samples from the MS COCO 2017 and PASCAL VOC datasets predominantly originate from real-world environments, which poses challenges for detecting less prominent objects that tend to be small in size or easily occluded. Comparative experiments evaluating CAM-YOLO against several contemporary methods for real-time object detection evaluated on two extensive publicly available datasets are presented in Tables 1 and 2.

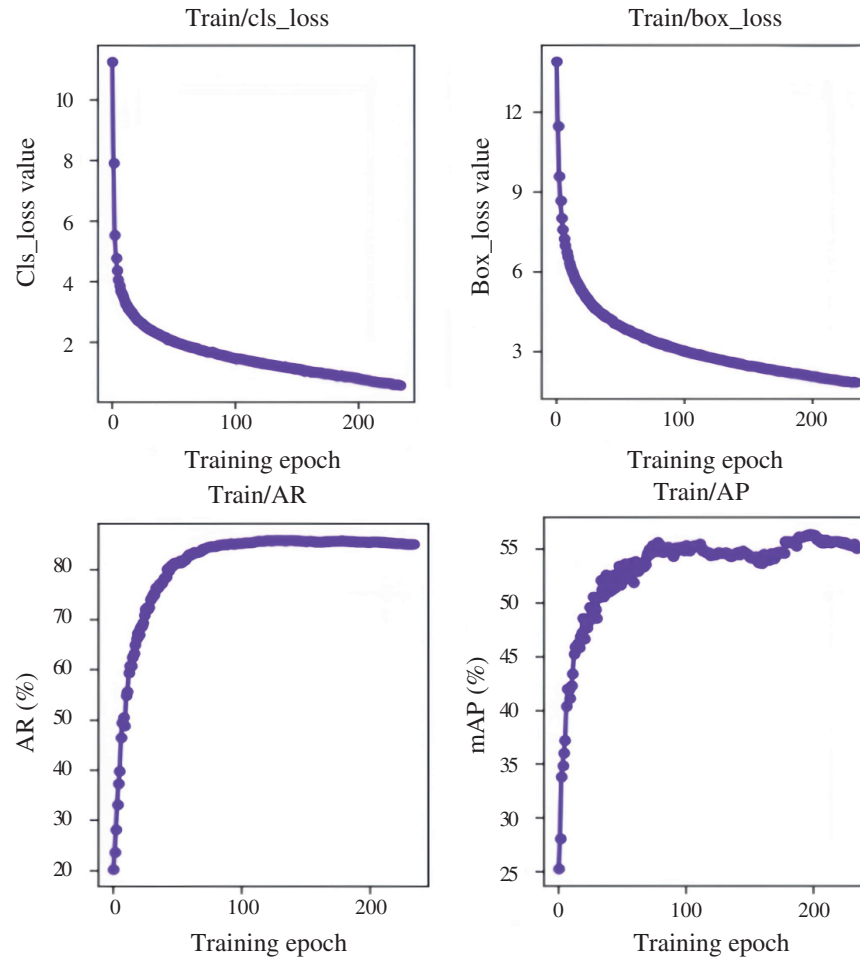


Figure 9: Fluctuations in selected metrics during the training process

Table 1: Benchmark evaluation of real-time detectors on COCO 2017 test set

Framework	mAP	AP^{50}	AP^{75}	AP^S	AP^M	FPS	$GFLOPs$
DETR [10]	43.3	59.5	46.4	24.1	48.5	76	224
YOLOv5x [12]	50.6	65.4	54.2	31.4	55.3	81	231
YOLOv6l [4]	52.8	69.2	58.1	35.4	55.2	91	221
YOLOv7x [5]	53.1	71.2	57.8	33.8	57.1	112	209
YOLOR [31]	55.4	73.3	60.6	38.7	58.6	31	245
PPYOLOE-X [32]	54.7	72.0	59.9	37.9	59.3	93	244
YOLOv8x [6]	53.9	72.6	58.5	36.1	57.3	134	188
YOLOv9-E [33]	55.4	72.7	60.5	36.9	58.6	138	233
DINO-DETR [34]	54.0	72.0	58.9	35.2	58.6	133	198
YOLO-MS [35]	54.2	72.2	59.0	36.2	58.0	114	207
Co-DETR [36]	54.3	71.9	59.2	35.6	59.2	138	203
RT-DETR-X [20]	54.8	73.1	59.4	35.7	59.6	146	189

(Continued)

Table 1 (continued)

Framework	mAP	AP^{50}	AP^{75}	AP^S	AP^M	FPS	$GFLOPs$
YOLOv10x [15]	54.3	73.0	60.1	37.2	58.4	149	161
YOLOv11x [37]	54.7	72.5	60.7	37.4	59.0	143	194
YOLOv12x [16]	55.1	73.2	60.5	37.6	59.5	137	199
YOLOv13x [17]	54.8	72.8	60.0	37.0	58.6	155	200
CAM-YOLO	56.7	75.8	61.1	41.2	59.8	120	176

Table 2: Findings from experiments conducted on the VOC 2007 test dataset

Framework	$mAP(IoU = 0.5)$	FPS	$GFLOPs$
RetinaNet [18]	85.2	65	198
YOLOv5x [12]	87.8	83	231
YOLOv6l [4]	89.9	94	221
YOLOv7x [5]	90.7	112	209
Deformable-DETR [11]	90.2	79	198
CBNetV2 [38]	90.9	28	215
YOLOR [31]	91.2	30	245
DAMO-YOLO [39]	89.3	126	226
DINO-DETR [34]	89.4	132	198
YOLO-MS [35]	89.9	114	208
Co-DETR [36]	90.3	137	204
YOLOv8x [6]	90.6	134	188
YOLOv9-E [33]	91.9	138	233
YOLOv10x [15]	91.5	148	161
YOLOv11x [37]	91.7	194	142
YOLOv12x [16]	92.1	137	199
YOLOv13x [17]	91.6	152	200
CAM-YOLO	93.1	121	176

The comparative experiments on the MS COCO 2017 test set corroborate that CAM-YOLO yields an overall precision of 56.7%, surpassing several well-established real-time object detection methods. Additionally, when compared with similar methodologies [5,6,12,15], CAM-YOLO demonstrated commendable performance across various precision and real-time metrics. Notably, in comparison to the reference framework (YOLOv8x), CAM-YOLO attained a 2.8% performance gain in accuracy. Furthermore, CAM-YOLO exhibited a significant 5.1% enhancement in the AP^S metric, which is beneficial for detecting less prominent targets in images.

Results according to the comparative experimental data obtained from Table 2 validate that CAM-YOLO shows a distinct advantage in precision relative to selected object detection methods. Specifically, it attains a 2.5% improvement over the baseline model. Although CAM-YOLO did not achieve optimal inference efficiency, its performance remains acceptable for most real-time scenarios.

4.3 Ablation Study

To evaluate the contribution of the proposed enhancement components, controlled ablation experiments were conducted on the MS COCO 2017 val set, comprising over 5000 images spanning diverse scenes and object categories. Individual modules were incrementally incorporated to quantify their specific effects on baseline capability.

4.3.1 Ablation Study of Backbone

This ablation study investigates the effects of the MSAM and CAM modules embedded within the foundational framework's detection results, with findings from the MS COCO 2017 val set recorded on [Table 3](#).

Table 3: Backbone enhancement effects

Baseline	MSAM	CAM	mAP	AP^S	AR
YOLOv8x	—	—	54.1	36.0	80.1
	✓	—	54.8	37.0	81.8
	—	✓	55.1	37.7	82.7
	✓	✓	55.5	38.5	84.2

The findings derived from the ablation experiments suggest that the incorporation of the MSAM and CAM components enhances the YOLOv8x model's performance in mAP, AP^S , and AR metrics by 1.4%, 2.5%, and 4.1%, respectively. MSAM effectively leverages multi-scale information fusion to enable deeper feature maps to capture more detailed information from the shallow layers, thereby facilitating better integration of hierarchical information across multiple depths. Additionally, the inclusion of CAM improves the modeling of cross-domain spatial feature relationships, further strengthening the backbone's representation of contextual semantic features.

4.3.2 Ablation Study of Attention Block

To provide additional evidence supporting the superiority of the proposed DBAM module, this segment explores the effect of augmenting the reference framework with various visual attention mechanisms on the accuracy metrics. The ablation study results are presented in [Table 4](#).

Table 4: Ablation study on attention block

Method	AP	AP^S	AR
YOLOv8x	54.1	36.0	80.1
YOLOv8x+CBAM [21]	54.9	37.0	81.0
YOLOv8x+GAM [22]	55.0	36.7	81.3
YOLOv8x+CA [23]	55.1	36.8	81.5
YOLOv8x+HSA [40]	55.0	37.1	81.2
YOLOv8x+EMA [41]	55.4	37.5	82.0
YOLOv8x+DBAM	55.7	38.9	81.9

The experimental findings demonstrate that DBAM surpasses alternative visual attention approaches in empowering the model to acquire more distinctive feature representations, consequently improving comprehensive detection performance. The integration of DBAM led to varying degrees of improvement across all three metrics, with the enhanced model achieving a 2.9% increase in AP^S , reflecting DBAM's optimization for detecting small-scale objects in images.

4.4 Visual Analytics

To delve deeper into the changes in the regions of interest during the decision-making process pre- and post-improvement, we utilized Grad-CAM to visualize several object detection models. The class activation maps illustrate the areas of the input images that contribute most significantly to the predictions of each class, as shown in Fig. 10.

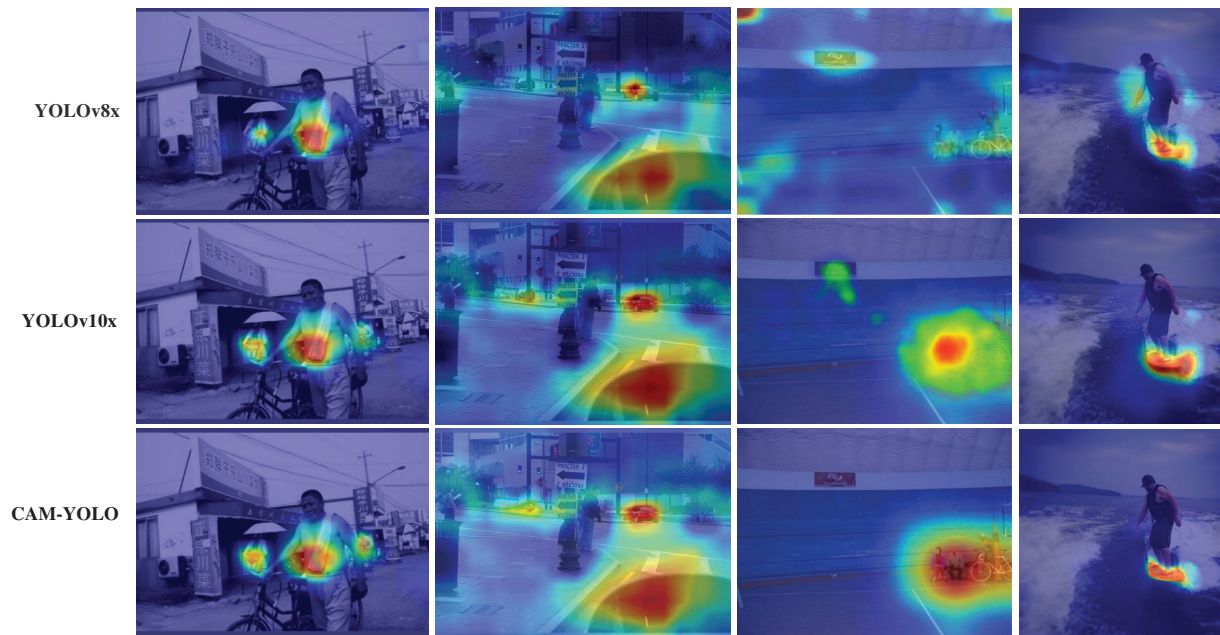


Figure 10: Class activation heatmaps [42] for various object detection models

The Grad-CAM results indicate that the CAM-YOLO model allocates greater attention to less prominent objects in images and demonstrates improved resilience to background noise interference.

The differences in detection performance between the baseline YOLOv8x model (top) and the proposed CAM-YOLO model (bottom) are illustrated in Fig. 11. Observing the detection results on identical image samples reveals a clear performance gap—why does this occur? The comparison highlights a limitation of the baseline YOLOv8x model: it often struggles to generate complete bounding boxes for non-salient targets, such as heavily occluded objects or small-scale instances, due to their inherently incomplete feature representation. This is precisely where our CAM-YOLO addresses the issue. Its architecture, which emphasizes multi-scale feature fusion and contextual modeling, enables accurate regression of these challenging targets. For instance, it successfully detects the occluded truck in the second image group and the small television in the third, cases where the baseline model faltered.



Figure 11: Detection performance differences on select images before and after model enhancements

5 Discussion

This study presents CAM-YOLO, which is inspired by the fundamental framework of YOLOv8x. Through optimizations in both the backbone architecture and information processing methods of YOLOv8, CAM-YOLO achieves more precise detection of less prominent targets within images. Specifically, the backbone of YOLOv8 employs extensive convolutional and pooling operations, which may contribute to the deterioration of detailed information pertinent to less noticeable objects. To mitigate this issue, the MSAM module has been introduced to assist the backbone network in preserving more shallow texture features, as illustrated in Table 3. Furthermore, to better compensate for the limitations in receptive field size and context representation introduced by the convolutional architecture, CAM has been embedded in the neck network to facilitate feature refinement of extended interdependencies within the spatial feature space, as shown in Table 3. Finally, our work proposes a novel visual attention mechanism, DBAM, which employs multi-dimensional weight modeling in both spatial and channel domains to augment the model's attention to vital characteristics found within images, with results demonstrated in Table 4.

CAM-YOLO exhibits impressive performance on the MS COCO 2017 and PASCAL VOC datasets. Relative to several classic real-time object detection Framework, CAM-YOLO demonstrates notable performance in precision and runtime, as illustrated in Tables 1 and 2. Nevertheless, it is important to highlight that the computational cost of CAM-YOLO has not yet achieved an ideal standard. This phenomenon stems mainly from the incorporation of supplementary network components and sophisticated feature integration schemes. While these enhancement modules have strengthened the model's detection proficiency, they concurrently introduce computational overhead that reduces inference velocity. Despite this, CAM-YOLO maintains a frame rate sufficient for predominant real-time applications.

6 Conclusion

This work proposes a fully integrated detection pipeline entitled CAM-YOLO anchored in the YOLOv8x architecture. Comparative experiments conducted on the MS COCO 2017 and PASCAL VOC datasets reveal the advantages of this method with respect to accuracy and runtime. By implementing enhancements in the architecture of the network and refining the techniques of information

processing employed in YOLOv8x, CAM-YOLO achieves notable breakthroughs in object detection accuracy. Future work will explore further architectural refinements to the YOLOv8 framework and assess its integration potential with state-of-the-art detection paradigms, including InternImage [43], YOLO-NAS [44], and YOLOv11 [37]. The proposed DBAM attention module will also be extended to adjacent vision tasks—such as image segmentation and object tracking—to comprehensively evaluate its scalability and generalization capabilities.

Acknowledgement: This work was supported by the Huangshan Technology Innovation Center for Digital Economy and Big Data Analysis.

Funding Statement: This work was primarily supported by the Talent Startup Program of Huangshan University under Grant No. 2025xkj003. Additional partial funding was gratefully received from the Scientific Research Project of the Anhui Provincial Department of Education under Grant No. 2025AHGXZK40303. We also gratefully acknowledge the partial support for this work from the Research Program of Study Abroad for Outstanding Young Scholars in Universities of Anhui Province (Grant No. gxgwx2018074).

Author Contributions: Chengang Dong performed the preliminary study and composed the outline of the paper and performed the experiments. Yongkang Ding composed the drawings and conducted the experiments, Jianwei Hu prepared the tables and read the proofread of the article. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: This study does not introduce new datasets. Instead, our work focused on methodological development and validation—an approach that relied entirely on established public benchmarks. Consequently, no original data require archiving.

Ethics Approval: As a computational study, this research exclusively analyzes publicly available benchmark datasets. Therefore, no specific ethics approval was required—the work involved no human participants, animal experimentation, or collection of private data.

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

References

1. Sun Z, Cao S, Yang Y, Kitani KM. Rethinking transformer-based set prediction for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct 10–17; Montreal, QC, Canada. p. 3611–20.
2. Ge Z, Liu S, Wang F, Li Z, Sun J. Yolox: exceeding yolo series in 2021. arXiv:2107.08430. 2021.
3. Wang H, Jin Y, Ke H, Zhang X. DDH-YOLOv5: improved YOLOv5 based on double IoU-aware decoupled head for object detection. J Real-Time Image Process. 2022;19(6):1023–33. doi:10.1007/s11554-022-01241-z.
4. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications. arXiv:2209.02976. 2022.
5. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 7464–75.
6. Aboah A, Wang B, Bagci U, Adu-Gyamfi Y. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 5349–57.

7. Feng C, Zhong Y, Gao Y, Scott MR, Huang W. TOOD: task-aligned one-stage object detection. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada: IEEE Computer Society. p. 3490–9.
8. Chowdhury PN, Sain A, Bhunia AK, Xiang T, Gryaditskaya Y, Song YZ. FS-COCO: towards understanding of freehand sketches of common objects in context. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2022. p. 253–70.
9. Tong K, Wu Y. Rethinking PASCAL-VOC and MS-COCO dataset for small object detection. J Vis Commun Image Represent. 2023;93(4):103830. doi:10.1016/j.jvcir.2023.103830.
10. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2020. p. 213–29.
11. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable DETR: deformable transformers for end-to-end object detection. arXiv:2010.04159. 2020.
12. Jocher G, Chaurasia A, Stoken A, Borovec J, Kwon Y, Michael K, et al. ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. Zenodo. 2022.
13. Pathak D, Raju U. Content-based image retrieval using group normalized-inception-darknet-53. Int J Multimed Inform Retrieval. 2021;10(3):155–70. doi:10.1007/s13735-021-00215-4.
14. Ding X, Zhang X, Ma N, Han J, Ding G, Sun J. RepVGG: making VGG-style convnets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 13733–42.
15. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. Yolov10: real-time end-to-end object detection. arXiv:2405.14458. 2024.
16. Tian Y, Ye Q, Doermann D. Yolov12: attention-centric real-time object detectors. arXiv:2502.12524. 2025.
17. Lei M, Li S, Wu Y, Hu H, Zhou Y, Zheng X, et al. YOLOv13: real-time object detection with hypergraph-enhanced adaptive visual perception. arXiv:2506.17733. 2025.
18. Wang Y, Wang C, Zhang H, Dong Y, Wei S. Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery. Remote Sens. 2019;11(5):531. doi:10.3390/rs11050531.
19. Dai X, Chen Y, Yang J, Zhang P, Yuan L, Zhang L. Dynamic DETR: end-to-end object detection with dynamic attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct 10–17; Montreal, QC, Canada. p. 2988–97.
20. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. Detrs beat YOLOs on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024 Jun 16–22; Seattle, WA, USA. p. 16965–74.
21. Woo S, Park J, Lee JY, Kweon IS. CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). Cham, Switzerland: Springer; 2018. p. 3–19.
22. Liu Y, Shao Z, Hoffmann N. Global attention mechanism: retain information to enhance channel-spatial interactions. arXiv:2112.05561. 2021.
23. Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 13713–22.
24. Qiu Y, Liu Y, Chen Y, Zhang J, Zhu J, Xu J. A2SPPNet: attentive atrous spatial pyramid pooling network for salient object detection. IEEE Trans Multimed. 2022;25:1991–2006.
25. Zhu L, She Q, Li D, Lu Y, Kang X, Hu J, et al. Unifying nonlocal blocks for neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct 10–17; Montreal, QC, Canada. p. 12292–301.
26. Han K, Xiao A, Wu E, Guo J, Xu C, Wang Y. Transformer in transformer. Adv Neural Inform Process Syst. 2021;34:15908–19.

27. Wang X, Girshick R, Gupta A, He K. Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 7794–803.
28. Dadboud F, Patel V, Mehta V, Bolic M, Mantegh I. Single-stage UAV detection and classification with YOLOV5: mosaic data augmentation and panet. In: 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS); 2021 Nov 16–19; Washington, DC, USA. p. 1–8.
29. Pan H, Guo Y, Yu M, Chen J. Enhanced long-tailed recognition with contrastive cutmix augmentation. *IEEE Trans Image Process.* 2024;33(4):4215–30. doi:10.1109/TIP.2024.3425148.
30. Li Q, Li D, Zhao K, Wang L, Wang K. State of health estimation of lithium-ion battery based on improved ant lion optimization and support vector regression. *J Energy Storage.* 2022;50(11):104215. doi:10.1016/j.est.2022.104215.
31. Chang HS, Wang CY, Wang RR, Chou G, Liao HY. YOLOR-based multi-task learning. arXiv:2309.16921. 2023.
32. Liu B, Bai X, Su X, Song C, Yao Z, Wei X, et al. DAC-PPYOLOE+: a lightweight real-time detection model for early apple leaf pests and diseases under complex background. In: 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC); 2023 Jun 26–30; Torino, Italy. p. 174–82.
33. Wang CY, Yeh IH, Mark Liao HY. YOLOV9: learning what you want to learn using programmable gradient information. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2025. p. 1–21.
34. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. Dino: DETR with improved denoising anchor boxes for end-to-end object detection. arXiv:2203.03605. 2022.
35. Xie Y, Zhang L, Yu X, Xie W. YOLO-MS: multispectral object detection via feature interaction and self-attention guided fusion. *IEEE Trans Cognit Develop Syst.* 2023;15(4):2132–43.
36. Zong Z, Song G, Liu Y. Detrs with collaborative hybrid assignments training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p. 6748–58.
37. Khanam R, Hussain M. Yolov11: an overview of the key architectural enhancements. arXiv:2410.17725. 2024.
38. Liang T, Chu X, Liu Y, Wang Y, Tang Z, Chu W, et al. CBNET: a composite backbone network architecture for object detection. *IEEE Trans Image Process.* 2022;31:6893–906.
39. Xu X, Jiang Y, Chen W, Huang Y, Zhang Y, Sun X. Damo-YOLO: a report on real-time object detection design. arXiv:2211.15444. 2022.
40. Ren Z, Zhou Y, Chen Y, Zhou R, Gao Y. Efficient human pose estimation by maximizing fusion and high-level spatial attention. In: Proceedings of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021); 2021 Dec 15–18; Jodhpur, India. p. 1–6.
41. Ouyang D, He S, Zhang G, Luo M, Guo H, Zhan J, et al. Efficient multi-scale attention module with cross-spatial learning. In: Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2023 Jun 4–10; Rhodes Island, Greece. p. 1–5.
42. Chattopadhyay A, Sarkar A, Howlader P, Balasubramanian VN. Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV); 2018 Mar 12–15; Lake Tahoe, NV, USA. p. 839–47.
43. Wang W, Dai J, Chen Z, Huang Z, Li Z, Zhu X, et al. Internimage: exploring large-scale vision foundation models with deformable convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 14408–19.
44. Terven J, Córdova-Esparza DM, Romero-González JA. A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. *Mach Learn Knowl Extrac.* 2023;5(4):1680–716. doi:10.3390/make5040083.