Complex & Intelligent Systems (2024) 10:5459–5473
https://doi.org/10.1007/s40747-024-01448-6

ORIGINAL ARTICLE

TA-YOLO: a lightweight small object detection model based on multi-dimensional trans-attention module for remote sensing images

Minze Li[1] · Yuling Chen[2] · Tao Zhang[1] · Wu Huang[1,3]

Received: 19 September 2023 / Accepted: 6 April 2024 / Published online: 8 May 2024

Abstract

Object detection plays a vital role in remote sensing applications. Although object detection has achieved proud results in natural images, these methods are difficult to be directly applied to remote sensing images. Remote sensing images often have complex backgrounds and small objects, which results in a highly unbalanced distribution of foreground and complex background information. In order to solve the above problems, this paper proposes a multi-head channel and spatial trans-attention (MCSTA) module, which performs remote pixel interaction from the channel and spatial dimensions respectively to complete the attention feature capture function. It is a plug-and-play module that can be easily embedded in any other natural image object detection convolutional neural network, making it quickly applicable to remote sensing images. First, in order to reduce computational complexity and improve feature richness, we use a special linear convolution to obtain three projection features instead of the simple matrix multiplication transformation in Transformer. Second, we obtain trans-attention maps in different dimensions in a manner similar to the self-attention mechanism to capture the interrelationships of features in channels and spaces. In this process, we use a multi-head mechanism to perform parallel operations to improve speed. Furthermore, in order to avoid large-scale matrix operations, we specially designed an attention blocking mode to reduce computer memory usage and increase operation speed. Finally, we embedded the trans-attention module into YOLOv8, added a new detection head and optimized the feature fusion method, thus designing a lightweight small object detection model named TA-YOLO for remote sensing images. It has fewer parameters than the benchmark model YOLOv8, and its mAP on the PASCAL VOC and VisDrone data sets increased by 1.3% and 6.2% respectively. The experimental results prove the powerful function of the trans-attention module and the excellent performance of TA-YOLO.

Keywords Small object detection · Remote sensing images · Transformer · YOLOv8 · Convolutional neural network

Introduction

Object detection is a pivotal research area within computer vision [1], finding applications in diverse domains like autonomous driving, intelligent monitoring, and remote sensing image analysis [2–6]. Nonetheless, challenges persist due to intricate backgrounds, small object sizes, and occlusions. As a result, small object detection has emerged as a significant and intricate focus within this field [7]. Recent years have witnessed remarkable progress in object detection through deep learning. While comprehensive datasets like PASCAL VOC [8] and MS COCO [9] have facilitated model development, a challenge persists. Many models excel overall but struggle with detecting and recognizing small objects. This limitation is particularly problematic in specialized contexts such as remote sensing, where UAV or satellite images present unique complexities like dense small objects against intricate backgrounds. Researchers in small object detection [10–14] are addressing this issue by focusing on remote sensing datasets. Although deeper networks offer better feature extraction, they come with a higher parameter count. Additionally, for small object tasks, higher image res-

✉ Wu Huang
huangwu@scu.edu.cn

Minze Li
liminze511323@gmail.com

[1] Chengdu Techman Sofeware Co., Ltd, Chengdu 610100, People's Republic of China

[2] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, People's Republic of China

[3] School of Computer Science, Sichuan University, Chengdu 610065, People's Republic of China

olution is crucial for accuracy. Yet, real-world deployment on edge devices requires both accuracy and real-time performance. Achieving small object detection using lightweight models poses a significant challenge, but its practical value and implications drive the pursuit. This study addresses these challenges by investigating small object detection methods on remote sensing datasets. Furthermore, we demonstrate the efficacy of our approach on PASCAL VOC, showcasing its performance not only for small objects but also for medium and large object detection.

Presently, the YOLO series [15–22] of methods shines among diverse object detection techniques, standing as the most widely employed models in the industry. Over time, the YOLO series has culminated in YOLOv8 [22], an apex model that excels in both speed and accuracy, revolutionizing industry-standard object detection practices. Although YOLOv8 demonstrates noteworthy prowess on the MS COCO dataset, it remains challenged in detecting small objects and struggles to directly apply its model to scenes teeming with such diminutive objects. Hence, our study builds upon the YOLOv8 architecture. To start, we observe that the detection heads responsible for loss calculation in YOLOv8 predominantly rely on deeply hierarchical features, which, due to network oversampling, may forfeit some local detail information, detrimentally affecting small object detection. We ingeniously leverage shallower features, introducing a novel detection head tailored for extremely small objects. This augmentation facilitates the network's fusion of shallow local detail information while preserving deep high-level semantic features. Additionally, the fundamental architecture of the Path Aggregation Network (PAN) [23] underpins YOLOv8, enabling feature transfer and fusion across different levels. However, recurrent upsampling and downsampling operations introduce variations in feature fusion across the same level, leading to variable degrees of information loss. In response, we harness the intrinsic features of this layer within the backbone and employ residual connections to mitigate lost features.

Crucially, small object detection invariably grapples with the intricate challenge of an imbalanced distribution between foreground and complex background information. Inspired by the self-attention mechanism within Transformers [24], we introduce a novel transformer-based multi-dimensional attention feature extraction method, thereby crafting the Multi-head Channel and Spatial Trans-Attention (MCSTA) module. This module performs remote pixel interaction from different dimensions to realize the attention feature capture function. It is a plug-and-play module that can be easily embedded in any other natural image object detection convolutional neural network, making it suitable for remote sensing images. In order to reduce computational complexity, reduce memory usage, and improve feature richness, a special linear convolution is incorporated into the module, a block attention mode is specially designed, and a multi-head mechanism is used for parallel operations. Based on the above design, we redesigned the network of YOLOv8 and proposed a lightweight small object detection model named TA-YOLO for remote sensing images.

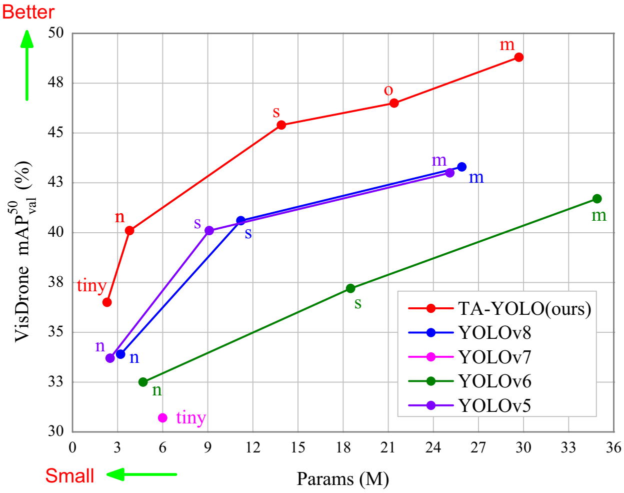

Fig. 1 Comparison of experimental results between TA-YOLO and other methods on the VisDrone validation set

Our contribution can be summarized as follows :

- In order to solve the problem of extremely unbalanced distribution of foreground and complex background information in remote sensing images, we designed a trans-attention module, which performs remote pixel interaction from two dimensions of channel and space to realize the attention feature capture function. It is also a plug-and-play module that can quickly apply other natural image object detection methods to remote sensing images.
- We used YOLOv8 as the baseline model and redesigned it. Including embedding the trans-attention module, adding a small object detection head, and optimizing the same-level feature fusion method. Therefore, a lightweight small object detection model named TA-YOLO for remote sensing images is proposed.
- Our TA-YOLO exhibits superior performance on the remote sensing image dataset VisDrone [25] with a large number of dense small objects, as shown in Fig. 1. Our method achieves higher accuracy with fewer parameters than YOLOv8. This is also a powerful performance of the proposed trans-attention module to apply the natural image object detection method to remote sensing images.

The remainder of the paper is organized as follows: "Related work" briefly describes related work. "Methodology" focuses on TA-YOLO, including multi-level feature fusion and MCSTA module. "Experiments" presents the data set used in the experiment, the experimental results, and the ablation experiments. Finally, "Conclusion" provides a summary and future prospects.

Related work

Object detection based on CNN

In recent years, the remarkable performance of convolutional neural networks (CNNs) in computer vision has driven extensive research in deep learning. CNN-based object detection methods can be broadly classified into two categories: two-stage and one-stage detection methods. The former employs a Region Proposal Network (RPN) to generate candidate object boxes, followed by classification and localization. Examples include RCNN [26], Fast R-CNN [27], Faster R-CNN [28], and Mask R-CNN [29], excelling in accuracy but with slower speeds and more parameters.

In contrast, one-stage detection methods directly detect objects in images, eliminating the need for candidate box generation. Notable methods include the YOLO series, MobileNet series [30–32], ShuffleNet series [33, 34], SSD [35], RetinaNet [36], and EfficientNet [37], known for real-time suitability. The YOLO series has garnered significant attention. YOLOv1 [15] pioneered one-stage detection using Grid Cells for bounding box and class prediction. Subsequent versions improved features, scales, and loss functions. YOLOv4 [18] introduced CSPDarknet53, Spatial Attention Module (SAM), PAN, and CIOU loss. YOLOv5 [19] refined Spatial Pyramid Pooling Fusion (SPPF), diverse training, and balanced loss weights. YOLOv6 [20] focused on efficiency with EfficientRep Backbone and Rep-PAN. YOLOv7 [21] introduced Extended Efficient Layer Aggregation Network (E-ELAN) and Reparameterized Convolution for feature extraction. YOLOv8 [22] advanced with the Compact Context Fusion (C2f) module, Diagonal-Free Loss (DFL) and CIOU Loss, and Task-Aligned Assigner. YOLOv8 is undoubtedly outstanding. However, it still has major limitations in the detection task of small objects.

Small object detection

Small object detection holds significant prominence and poses substantial challenges in the field of computer vision. Remote sensing image data is commonly employed to evaluate the performance of small object detection. Such images often encompass diverse land features, making the targets susceptible to interference from various similar features like color, texture, and shape. Hu et al. [38] utilized a coarse image pyramid and employed twice upsampled input images for detecting small faces. Zheng et al. [39] developed a multi-receptive field convolutional group representation aggregation module to expand the receptive field, thoroughly capturing both the intrinsic features of targets and the semantic relations with the surrounding environment, thereby enhancing detection performance. Liu et al. [40] increased the number of small object training examples by downsizing large objects. D-SSD [41], C-SSD [42], F-SSD [43], and ION [44] focused on constructing suitable context features for small object detection. Furthermore, there are studies utilizing Generative Adversarial Networks (GANs) to generate super-resolution features for small object detection, such as [45] and [46]. However, larger input resolutions and super-resolution methods incur higher computational costs, rendering them unsuitable for lightweight detector designs. Given the nature of small objects, networks often necessitate intricate designs to effectively capture minute features. The aforementioned methods also encounter similar issues of computational complexity. We observe a scarcity of attention devoted to utilizing lightweight object detection frameworks for small object detection tasks. In this paper, our objective is accurate small object detection while maintaining lower computational complexity.

Attention mechanism

The Attention Mechanism enhances a model's focus on specific information or regions by adaptively assigning weights to input elements. It finds application in tasks across machine learning, such as natural language processing, computer vision, and speech recognition. Addressing unbalanced foreground-background distribution in object detection, especially for small objects, highlights the significance of attention mechanisms.

The Recurrent Attention Model (RAM) [47] integrates attention with deep neural networks. It iteratively selects and emphasizes image regions, enhancing accuracy and efficiency. SENet [48] introduces attention to convolutional neural networks, learning channel importance with Squeeze-and-Excitation modules. CBAM [49] combines channel and spatial attention for feature enhancement, adapting to diverse image features. Coordinate Attention [50] adds spatial relationship consideration to attention mechanisms, weighting spatial positions for task relevance. Originally in Transformers, the self-attention mechanism computes Query, Key, and Value similarity-based weights for weighted pooling, capturing global dependencies for long-range associations. Inspired by this, we adopt its weight distribution and Multi-Head mechanism to channel and spatial attention extraction.

Methodology

Revisiting transformer

Transformer is a network based on self-attention mechanisms. The input consists of query and key of dimension d_k, and value of dimension d_v. Multi-head attention is a mechanism that involves employing multiple distinct, learned linear transformations to project the query, key, and value vectors h times onto separate dimensions denoted as d_q, d_k, and d_v. Subsequently, attention operations are conducted in parallel on each of these transformed versions of queries, keys, and values. This orchestrated process results in h sets of output values, each encompassing d_v dimensions. Transformer architecture further enhances this by amalgamating the outcomes of different attention heads using learnable weights, thereby fostering an adaptive linear aggregation of information. Let q indexes a query element with feature $x_q \in \mathbb{R}^C$, k indexes a key element with feature $x_k \in \mathbb{R}^C$, and v indexes a value element with feature $x_v \in \mathbb{R}^C$, where C is the feature dimension. During the actual operational phase, we simultaneously compute the attention function on a set of queries, packed them into a matrix Q. The keys and values are packed into matrices K and V in the same way. Then they can be expressed as $X_q \in \mathbb{R}^{d_q}$,$X_k \in \mathbb{R}^{d_k}$,and $X_v \in \mathbb{R}^{d_v}$ after being projected h times. Then the multi-head attention feature is calculated by

$$\begin{aligned} &MultiHead(X_q, X_k, X_v) \\ &\quad = W^O \mathbb{C}_{i=1}^{h} Attention_i(X_q, X_k, X_v), \end{aligned} \tag{1}$$

$$\begin{aligned} &Attention_i(X_q, X_k, X_v) \\ &\quad = W_i^V X_v Softmax\left(\frac{W_i^Q X_q (W_i^K X_k)^{\mathrm{T}}}{\sqrt{d_k}}\right), \end{aligned} \tag{2}$$

where $\mathbb{C}$ means matrix concatenation. The projections are parameter matrices $W_i^Q \in \mathbb{R}^{C \times d_q}$, $W_i^K \in \mathbb{R}^{C \times d_k}$, $W_i^V \in \mathbb{R}^{C \times d_v}$, and $W_i^O \in \mathbb{R}^{hd_v \times C}$. h indexes parallel attention heads and $d_q = d_k = d_v = C/h$.

TA-YOLO

The overall structure of TA-YOLO proposed in this paper is shown in Fig. 2. We have closely followed the architectural paradigm of YOLOv8, which similarly comprises three core components: the backbone, neck, and head. In our approach, the notations P2, P3, P4, P5 denote the feature map levels resulting from downsampling the input image by factors of 4, 8, 16, 32 respectively. These levels embody varying degrees of feature map granularity. Much like YOLOv8, our TA-YOLO encompasses a range of models with diverse scales, governed by the predetermined parameters w and n as depicted in the figure. At the termination of the backbone, we have seamlessly integrated the novel MCSTA module to bolster the fundamental feature extractor's capacity to capture foreground features effectively. Concurrently, this module has been strategically incorporated at the conclusion of each feature level within the neck. This strategic placement serves to further disentangle the foreground and background aspects of the final feature representation.

To aptly address the detection of exceedingly diminutive objects within remote sensing images, we have harnessed the higher-resolution features from the P2 level and introduced a specialized tiny object detection head. Furthermore, we've implemented feature fusion between the original P3 and P4 layers in the backbone. This serves to counteract excessive information loss stemming from the iterative upsampling and downsampling processes. By infusing our pioneering MCSTA module and optimizing the network structure, TA-YOLO delivers commendable results across various object sizes. Its efficacy extends beyond medium and large objects, exhibiting superior performance in detecting small and even minuscule objects. This comprehensive design augmentation equips TA-YOLO to excel across a spectrum of object detection scenarios.

Improvement of multi-level feature fusion

The significance of multi-level feature fusion in small object detection cannot be understated. In YOLOv8, object detection relies solely on features from three levels: P3, P4, and P5. However, when dealing with extremely small and densely packed objects within remote sensing images, this approach faces challenges. Features at the P2 level exhibit a higher resolution, offering the network a valuable opportunity to capture intricate texture details present in the image. Building upon this insight, we have introduced a specialized detection head aimed at detecting tiny objects. This innovative addition, depicted by the red dashed box in Fig. 2, enhances our model's ability to identify minuscule objects.

In the neck architecture, we have integrated the concept of PAN, which operates through a sequence of Down-Up-Down sampling (D-U-D) steps. In YOLOv8, the fusion of features during the second round of D-U-D is limited to the same-level upsampling process. While this helps mitigate information loss to some extent, it remains insufficient for particularly small objects. To address this, we leverage the original features from each level in the backbone. Utilizing residual connections, we compensate for any lost features resulting from repeated upsampling and downsampling processes. Fig. 2 illustrates this approach with red arrows. Notably, the Concat operation at the P5 level amalgamates features processed via SPPF. However, these features remain devoid of repeated upsampling or downsampling. This strategic decision preserves the unique characteristics of these features,

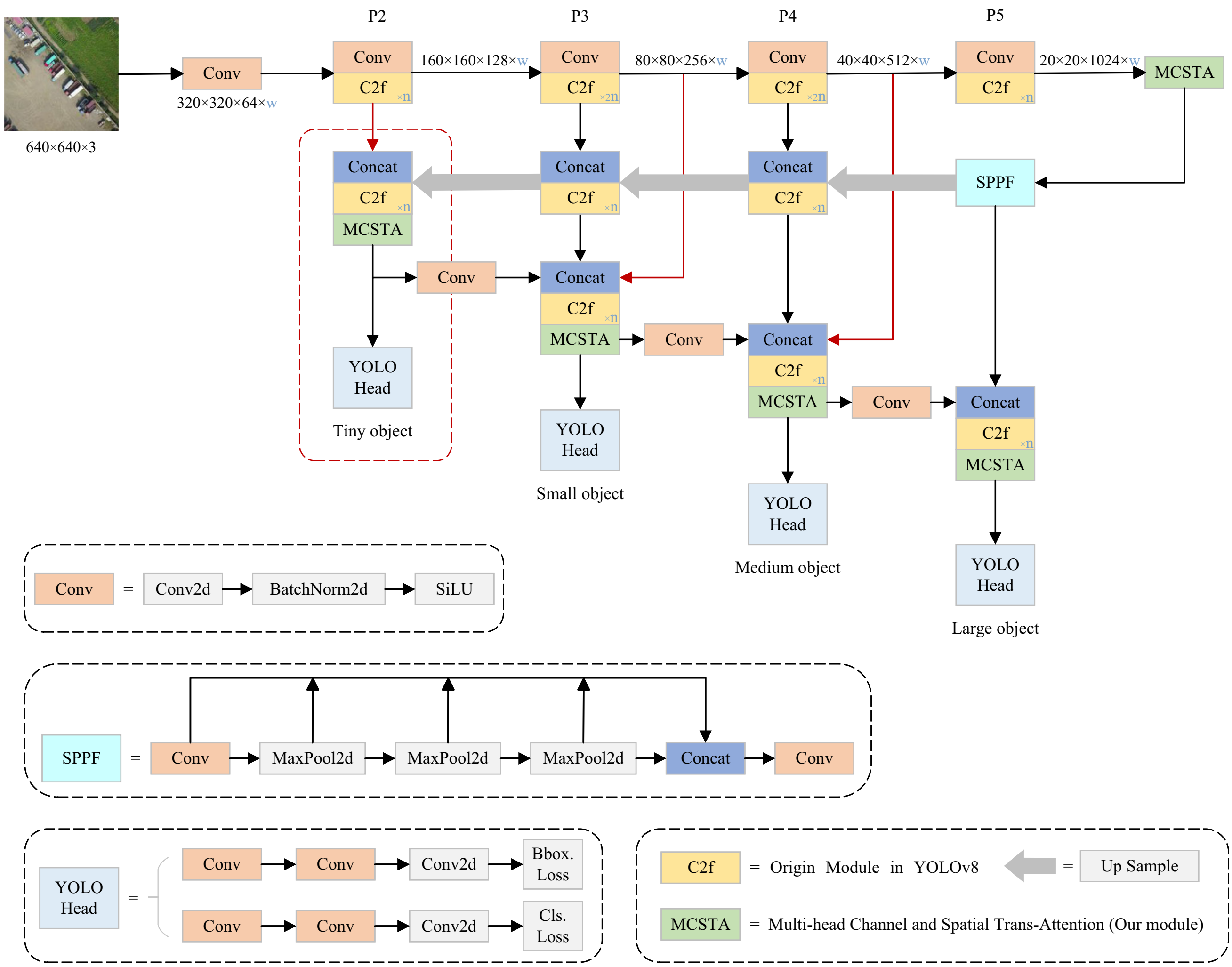

Fig. 2 The overall structure of TA-YOLO. The red dotted box and arrow represent the new detection head and feature fusion method added compared to YOLOv8, respectively. w and n are used to control the size of the network

preventing excessive fusion and subsequently promoting a more lightweight model architecture.

Through these refined design choices, our model excels in small object detection, benefiting from a comprehensive multi-level feature fusion strategy that harnesses both high-resolution features and precise information propagation.

MCSTA module

In the context of small object detection, grappling with the highly imbalanced distribution between foreground and complex background information becomes an unavoidable challenge, particularly pronounced in remote sensing images. This discrepancy presents a formidable obstacle for networks to accurately discern the intricate details of exceedingly small objects within vast-scale images.

To address this issue, we have innovated the MCSTA module. This module serves a dual purpose: firstly, it enhances the extraction of channel trans-attention features via the MCTA submodule; secondly, it proceeds to extract spatial trans-attention features through the MSTA submodule. This sequential process culminates in the fusion of these extracted features with the initial input, yielding a refined and optimized feature representation. The MCSTA module's architectural blueprint is illustrated in Fig. 3, consisting of the aforementioned MCTA and MSTA sub-modules. To succinctly outline the MCSTA process: consider an input denoted as Z, and the ensuing output as $Z^{'}$. The MCSTA module can be summarized as follows:

$$Z^{'} = Z + W_2^{msta}(W_1^{mcta} Z), \tag{3}$$

where $\{W_1^{mcta}, W_2^{msta}\} \in \mathbb{R}^{H \times W \times C}$ represent the weight parameters of the MCTA and MSTA modules, respectively. The design of the residual structure can ensure that our MSCTA module will not interfere with the follow-up results due to attention deviation or confusion during the feature extraction process, and it also plays a supervisory role. By enacting this sophisticated module, we bolster our model's ability to discern and highlight salient foreground features amidst complex background contexts, thereby elevating its proficiency in detecting small objects within the challenging realm of remote sensing imagery.

The inception of the MCTA and MSTA modules is deeply rooted in the conceptual framework of self-attention, a hallmark of the Transformer architecture. In a parallel trajectory, we retain the fundamental components of query, key, and value from Transformer, harnessing their power to enable information focus and extraction. It's important to note that while our inspiration draws from Transformer, we have tailored the design to cater to the unique attributes of image data. Conventionally, the Transformer's architecture divides images into standardized patches, each subsequently encoded with positional information for further processing. In a departure from this approach, we've steered clear of employing a block design. Instead, our module operates on the entire image. This modification not only simplifies the model's complexity but also obviates the need for explicit positional encoding. This streamlined strategy enhances efficiency without compromising performance.

Our module's adaptation of query, key, and value generation is distinct from Transformer's methodology. We derive these elements utilizing Ghost convolution [51], an exceptionally lightweight convolutional technique. Ghost convolution leverages linear transformations to eliminate feature redundancies, yielding enriched features with minimal computational overhead. This streamlined procedure, often referred to as a "cheap" operation, omits batch normalization and non-linear activations. This design choice optimally integrates ghost convolutions, emphasizing localized contextual details, culminating in the computation of feature covariances, which in turn generate comprehensive global attention maps.

Intrinsically linked, the MCTA and MSTA modules serve as embodiments of this adapted self-attention principle. Through carefully orchestrated operations, these modules deftly capture nuanced contextual cues, a subject we'll delve into further in the following elaboration.

MCTA module

As shown in Fig. 3, assuming an input feature $X \in \mathbb{R}^{H \times W \times C}$ is given, it is generated through three Ghost convolutions to generate query(Q), key(K), and value(V) projections, where $\{Q, K, V\} \in \mathbb{R}^{H \times W \times C}$. Then we reshape them to $\mathbb{R}^{N \times C}$, where $N = H \times W$ is the number of pixels. After that we perform matrix multiplication between the transpose of Q and K, and apply a softmax function to obtain the channel trans-attention map $\mathbf{A} \in \mathbb{R}^{C \times C}$:

$$a_{ji} = \frac{exp(Q_i \cdot K_j)}{\sum_{i=1}^{C} exp(Q_i \cdot K_j)}, \tag{4}$$

where a_{ji} measures the i^{th} channel's impact on the j^{th} channel. The more similar the feature representations of two channels are, the greater the correlation between them. Then, we perform matrix multiplication between features A and V, and reshape the result into $\mathbb{R}^{H \times W \times C}$. In order to speed up model convergence, similar to the Transformer structure, we also introduce Layer Normalization. The difference is that our Layer Normalization is after the attention feature extraction, not before. This effect will be better, which has been proven in Swin Transformer V2. To expedite convergence, akin to Transformers, we incorporate Layer Normalization. However, we place Layer Normalization post-attention feature extraction, a distinction that has proven notably effective in Swin Transformer V2 [52]. Finally, we perform an element-wise sum operation with X to obtain the final output $X' \in \mathbb{R}^{H \times W \times C}$:

$$X_j^{'} = X_j + \sum_{i=1}^{C} (a_{ji} X_i), \tag{5}$$

where $X_j^{'}$ represents the features of the j^{th} channel of $X^{'}$. Eq. 5 shows that the final feature of each channel is the weighted sum of the features of all channels and the original features, which illustrates the long-range semantic dependencies between feature maps of different channels. Similar to the multi-head self-attention in Transformer, we divide the number of channels into 'heads' and compute the respective attention maps in parallel. In practice, we realize it by reshaping Q, K and V to $\mathbb{R}^{H \times W \times C^{'} \times h}$, where $h = C/C^{'}$ represent the heads. Overall, the MCTA process is defined as:

$$X^{'} = X + Norm[\mathbb{C}_{i=1}^{h} Attention_i(X_q, X_k, X_v)], \tag{6}$$

$$Attention_i(X_q, X_k, X_v) = W_i^V X_v Softmax\left(\frac{(W_i^Q X_q))^{\mathrm{T}} W_i^K X_k}{\alpha}\right), \tag{7}$$

where $\{W_i^Q, W_i^K, W_i^V\} \in \mathbb{R}^{C \times C}$ respectively represent the weight parameters of Ghost convolution on the three branches. $X_q = X_k = X_v = X$ is the different representation of the input features on the three branches. Here, α is a learnable scaling parameter used to control the size of the Q and K dot product before applying the softmax function.

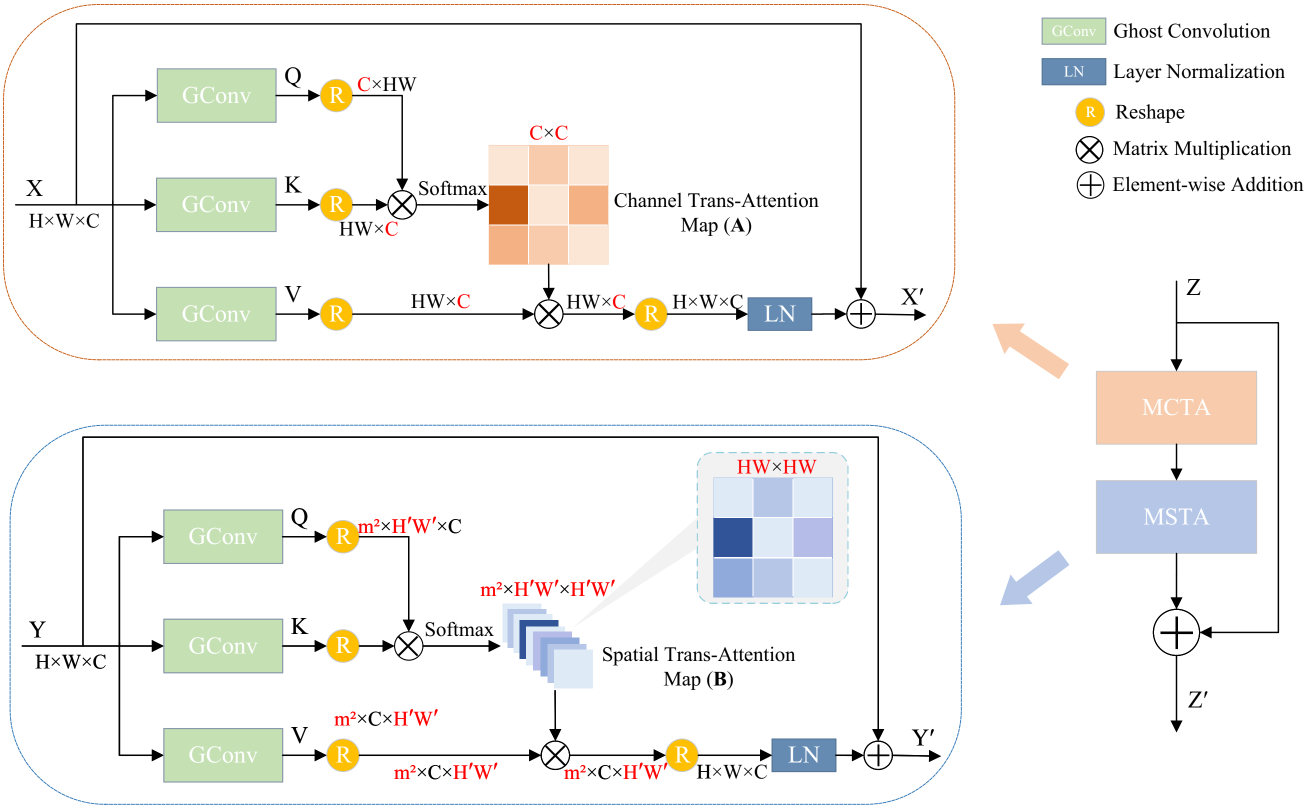

Fig. 3 Proposed MCSTA module. The two sub-modules extract channel trans-attention and spatial trans-attention respectively

MSTA module

MSTA follows a structure akin to MCTA, though with slight variations in local details. When provided with an input feature $Y \in \mathbb{R}^{H \times W \times C}$, the Q, K, and V projections are reshaped to $Y \in \mathbb{R}^{C \times N}$, where $N = H \times W$. A spatial trans-attention map $\mathbf{B} \in \mathbb{R}^{N \times N}$ is generated using a similar operation. However, this approach comes at a considerable cost. Multiplying the Q and K matrices results in a computational complexity of $O(H^2W^2C)$, growing quadratically with spatial size. For instance, given an input feature map of size $H = W = 100$, $\mathbf{B}$ becomes a sizable matrix $\mathbb{R}^{10^4 \times 10^4}$. This poses memory and speed challenges when embedding the module into shallow features, as evidenced by out-of-memory issues even with our minimum parameter TA-YOLO model on an NVIDIA GTX4090 GPU, even at batch size 1.

To address this, we segment the feature map into blocks and compute attention within each block separately. Notably, blocks cannot be too small, as excessively small blocks limit attention capacity. Our TA-YOLO design carefully customizes block divisions at each feature level, mapping each block's receptive field to encompass small and medium-sized objects. This enhances foreground information extraction for small objects. Assuming m^2 blocks divide the input feature map, the Q and K matrix multiplication complexity becomes $O(m^2(H/m)^2(W/m)^2C)$, substantially reducing computations. Furthermore, when $H = W = 100$ and $m = 10$, $\mathbf{B} \in \mathbb{R}^{m^2 \times 10^2 \times 10^2}$. Smaller matrices lend themselves to efficient parallelization, capitalizing on optimized parallel computing capabilities present in hardware devices like GPUs, thereby enhancing performance. After that, $\mathbf{B} \in \mathbb{R}^{m^2 \times N' \times N'}$:

$$b_{ji} = \frac{exp(Q_i \cdot K_j)}{\sum_{i=1}^{N'} exp(Q_i \cdot K_j)}, \tag{8}$$

where b_{ji} measures the i^{th} position's impact on the j^{th} position. Here $N' = HW/m^2$, and m^2 represents the number of blocks. Like MCTA, we will get the final output $Y' \in \mathbb{R}^{H \times W \times C}$:

$$Y'_j = Y_j + \sum_{i=1}^{N'} (b_{ji} Y_i), \tag{9}$$

where Y'_j represents the feature at the j^{th} position of Y'. Eq. 9 shows that the final feature at each position is the weighted sum of the features across all positions in the block and original features. Therefore, it has a block-like view of the global context and selectively captures contexts according to the spatial trans-attention map. Its heads design and overall process are kept consistent with MCTA, so we won't go into details here.

Experiments

To thoroughly assess our proposed approach, we undertake comprehensive experimentation on two distinct datasets: VisDrone [25] and PASCAL VOC [8]. The former encompasses drone vision data, abundant in small objects, while the latter serves as an internationally recognized benchmark for object detection, predominantly focusing on medium and large objects. Our experimental findings reveal that our model excels not only in detecting medium and large objects, but also demonstrates remarkable efficacy in small object detection.

Datasets

The VisDrone dataset [25] was introduced by the AISKYEYE team from Tianjin University's Machine Learning and Data Mining Laboratory in China. Comprising 288 video clips, this dataset encompasses 261,908 frames and 10,209 still images. It features a diverse range of scenarios and conditions, including different locations, environments, objects, densities, weather, and lighting conditions, captured using multiple drones for varied scenes and missions. The dataset is enriched with 10 categories, containing 6,471 training data, 548 validation data, and 3190 test data samples. Out of these, annotations are available for 1,610 test data samples. In contrast to conventional object detection datasets, the VisDrone dataset often includes hundreds of small objects within a single image, resulting in a substantial total of 2.6 million annotation boxes. Additionally, the dataset provides significant attributes like scene visibility, object class, and occlusion, enhancing its utility for diverse tasks.

The PASCAL VOC dataset [8], originally developed by the Computer Vision group at the University of Oxford, UK, is a well-known and widely employed general-purpose dataset in the field of computer vision. Primarily designed for tasks such as object detection, image segmentation, and image classification, it encompasses 20 common object categories. Spanning multiple years, the VOC dataset was created for the PASCAL VOC challenge. Each year's dataset contains separate training, validation, and test sets. Notably, data from the years 2007 and 2012 are frequently used. In contemporary practice, the training and validation data from both years are often combined into a unified training set, totaling 16,551 samples. Meanwhile, the test data from the year 2007 serves as the validation or test set, comprising a total of 4952 samples.

Table 1 Comparison of the instance size distributions of two datasets

Datasets	10–50 Pixels	50–300 Pixels	>300 Pixels
VisDrone	0.74	0.26	0
PASCAL VOC	0.14	0.61	0.25

The datasets exhibit notable differences in difficulty stemming from their distinct data compositions. Table 1 illustrates this contrast, indicating that 74% of the VisDrone dataset comprises small objects, whereas the PASCAL VOC dataset predominantly contains medium and large objects, accounting for 86%. To offer a visual representation of this divergence, Fig. 4 showcases selected images from both datasets.

Implementation details

The dataset division approach aligns with the methodology outlined in Datasets. In our experimentation, we have adopted several widely recognized evaluation metrics, namely Precision (P), Recall (R), mean Average Precision (mAP), model parameters (Params) and Giga Floating Point Operations Per Second (GFLOPs) [9]. The mean Average Precision (mAP) calculates the average of the average precision (AP) across all categories. This calculation involves utilizing an Intersection over Union (IOU) threshold, with a threshold above which indicating successful detection. We denote the results achieved at an IOU threshold of 0.5 as mAP^{50}. By varying the threshold from 0.5 to 0.95 with an incremental step of 0.05, we compute the average of these values to obtain $mAP^{50:95}$ [9]. Our implementation is based on PyTorch, with input sample sizes normalized to 640 × 640. The training employs the SGD optimizer with an initial learning rate of 0.01, a momentum of 0.937, and a weight decay of 0.0005 to prevent overfitting. We set the batch size to 16 and the maximum epochs to 500, incorporating early stopping with a patience of 50. Remaining hyperparameters are kept consistent with the default YOLOv8 settings. In order to ensure a fair evaluation, all comparative and ablation experiments exclude the use of pre-trained weights during training. Four crucial parameters underpin our TA-YOLO architecture: h, m, n, and w. Here, h represents the heads in the multi-head mechanism, with distinct values 1, 2, 4, 8 considered at each P2, P3, P4, P5 level. $m = 8/h$, its square dictating the number of blocks in the MSTA. The parameter n signifies the number of C2f module stacks, set at 1 for our study. Lastly, w governs the generated model's size, categorized into five sizes tiny, n, s, o, m, each aligned with w values 0.1875, 0.25, 0.50, 0.652, 0.75, respectively. The loss function undergoes optimization, incorporating DFL Loss in addition to CIOU Loss for enhanced positioning. The comprehensive loss function as follows:

$$Loss(all) = \lambda_1 Loss(bce) + \lambda_2 Loss(ciou) + \lambda_3 Loss(dfl), \quad (10)$$

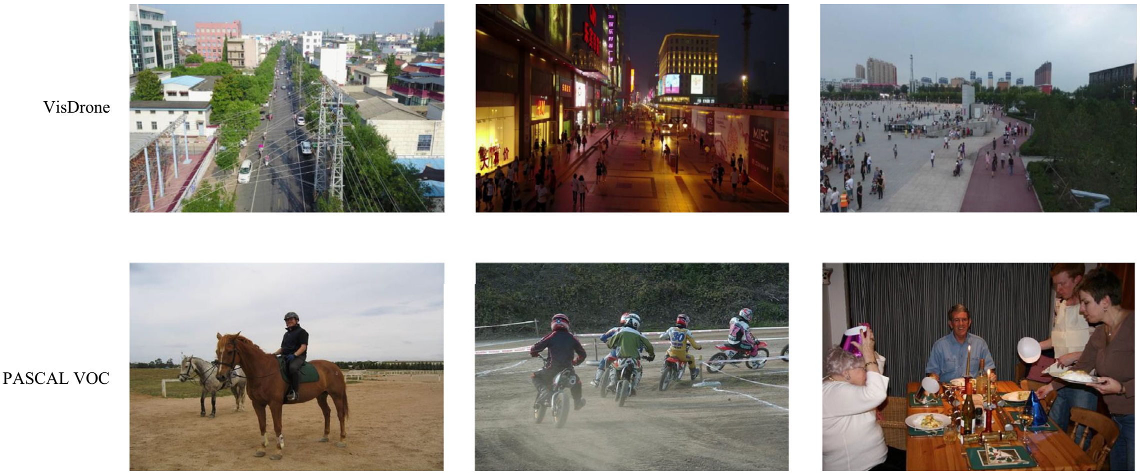

Fig. 4 Samples example of the two datasets

where $Loss(bce)$ corresponds to the classification loss function, which encompasses Binary Cross-Entropy (BCE) Loss. On the other hand, $Loss(ciou)$ and $Loss(dfl)$ jointly contribute to form the localization loss function. The parameter λ signifies a weighting factor used to quantify the loss. Notably, specific constants, namely λ_1, λ_2, and λ_3, are assigned values of 0.5, 7.5, and 1.5, respectively, as per YOLOv8's configuration.

Experimental results

Our primary objective is to design a small object detection model that excels in both speed and accuracy. Given this focus, our experimental comparisons exclude larger models. In the YOLO series, models are typically categorized into five different sizes: n, s, m, l, x. The parameter volume of the YOLOv8-l model has reached 43.7 million (M), so we restrict our analysis to the first three sizes: n, s, m. To offer a comprehensive performance assessment, we introduce two additional small-scale models, resulting in a total of five sizes for our TA-YOLO model: tiny, n, s, o, m.

Our model's architecture builds upon YOLOv8, prompting a thorough comparison with each size variant of YOLOv8 across the two datasets, as detailed in Tables 2, 3, and 4. Observing these tables reveals notable enhancements in the performance of our TA-YOLO model, achieved with a relatively modest increase in parameters. By comparing the tiny and o-sized models with the n and m sizes-where the parameters are closely matched-it becomes evident that TA-YOLO outperforms YOLOv8 while utilizing fewer parameters. This optimization justifies our augmentation of the tiny and o sizes.

Notably, TA-YOLO boasts superior speed and performance compared to YOLOv8, which is visually demonstrated in Fig. 1. On the PASCAL VOC dataset, our model exhibits improvements of up to 1.3%. However, when evaluated on the validation and test sets of VisDrone, our model achieves remarkable enhancements of up to 6.2% and 4.3% respectively. These findings underscore the exceptional proficiency of our proposed TA-YOLO in small object detection scenarios.

We have extended our comparative analysis beyond YOLOv8 and evaluated our model against other existing methods. As our research emphasizes small object detection, we exclusively employed the VisDrone dataset for these comparisons. The results of these comparative experiments are summarized in Table 5, which highlights the performance of various methods. Notably, our model achieves a compelling combination of minimal parameter quantity and superior accuracy.

Within this context, it's worth noting that, in addition to the YOLOv7-tiny model, YOLOv7 stands out as having the second lowest parameter count at 36.9M. Consequently, our comparison focuses solely on the YOLOv7-tiny model from the v7 series.

For a more visually intuitive presentation of the comparison results, we've crafted Fig. 1, which juxtaposes the two indicators- mAP^{50} and Params-from Table 5. This graphic depiction offers a concise yet informative representation of how our model outperforms other methods.

The depicted experimental outcomes aptly underscore TA-YOLO's commendable performance. In our pursuit to delve into our model's specific performance within the realm of small object detection, we present the Precision-Recall curves for each category on the VisDrone validation set, as vividly illustrated in Fig. 5. These curves manifest the outcomes achieved by setting the IOU threshold to 0.5, subsequently yielding mAP through the integral area under each curve.

Table 2 Comparison of experimental results on PASCAL VOC

Methods	P	R	mAP^{50}	$mAP^{50:95}$	Params (M)	GFLOPs
YOLOv8-n [22]	0.810	0.733	0.806	0.599	3.2	8.9
YOLOv8-s [22]	0.818	0.756	0.833	0.634	11.2	28.8
YOLOv8-m [22]	0.818	0.790	0.854	0.670	25.9	79.3
TA-YOLO-tiny	0.782	0.743	0.807 (↑0.1%)	0.608	2.3 (↓0.9)	7.9 (↓1.0)
TA-YOLO-n	0.817	0.742	0.820 (↑1.3%)	0.624	3.8 (↑0.6)	14.1 (↑5.2)
TA-YOLO-s	0.836	0.774	0.845 (↑1.2%)	0.656	13.9 (↑2.7)	43.3 (↑14.5)
YA-YOLO-o	0.827	0.787	0.854 (↑0%)	0.663	21.4 (↓4.5)	64.6 (↓14.7)
TA-YOLO-m	0.838	0.795	0.862 (↑0.8%)	0.680	29.7 (↑3.8)	110.2 (↑30.9)

Table 3 Comparison of experimental results on VisDrone validation set

Methods	P	R	mAP^{50}	$mAP^{50:95}$	Params (M)	GFLOPs
YOLOv8-n [22]	0.450	0.338	0.339	0.196	3.2	8.9
YOLOv8-s [22]	0.528	0.386	0.406	0.242	11.2	28.8
YOLOv8-m [22]	0.556	0.416	0.433	0.265	25.9	79.3
TA-YOLO-tiny	0.485	0.349	0.365 (↑2.6%)	0.218	2.3 (↓0.9)	7.9 (↓1.0)
TA-YOLO-n	0.502	0.389	0.401 (↑6.2%)	0.241	3.8 (↑0.6)	14.1 (↑5.2)
TA-YOLO-s	0.539	0.443	0.454 (↑4.8%)	0.277	13.9 (↑2.7)	43.3 (↑14.5)
YA-YOLO-o	0.558	0.450	0.465 (↑3.2%)	0.286	21.4 (↓4.5)	64.6 (↓14.7)
TA-YOLO-m	0.583	0.466	0.488 (↑5.5%)	0.302	29.7 (↑3.8)	110.2 (↑30.9)

Table 4 Comparison of experimental results on VisDrone test set

Methods	P	R	mAP^{50}	$mAP^{50:95}$	Params (M)	GFLOPs
YOLOv8-n [22]	0.405	0.301	0.279	0.158	3.2	8.9
YOLOv8-s [22]	0.456	0.347	0.329	0.190	11.2	28.8
YOLOv8-m [22]	0.487	0.369	0.353	0.207	25.9	79.3
TA-YOLO-tiny	0.419	0.307	0.292 (↑1.3%)	0.165	2.3 (↓0.9)	7.9 (↓1.0)
TA-YOLO-n	0.427	0.341	0.316 (↑3.7%)	0.180	3.8 (↑0.6)	14.1 (↑5.2)
TA-YOLO-s	0.488	0.379	0.371 (↑4.2%)	0.214	13.9 (↑2.7)	43.3 (↑14.5)
YA-YOLO-o	0.501	0.388	0.377 (↑2.4%)	0.219	21.4 (↓4.5)	64.6 (↓14.7)
TA-YOLO-m	0.521	0.399	0.396 (↑4.3%)	0.231	29.7 (↑3.8)	110.2 (↑30.9)

Evidently, “car” emerges as the standout performer within this ensemble, capturing the limelight with the highest accuracy. This is particularly noteworthy as “car” claims a substantial share of the dataset. On the contrary, the category with the least representation-namely “Awning-tricycle”-experiences the most challenging detection scenario due to obstructions caused by aerial views and overhanging structures. Consequently, its accuracy is notably lower. The “bicycle” category, with a relatively meager presence, presents further complexities. This challenge arises from occlusions caused by riders and the thin, elongated nature of bicycles at a distance. These factors diminish the availability of discernible and significant features, especially when juxtaposed against the prominence of people within the image. A visual glimpse into these complexities is provided in the example image of Fig. 4.

To provide a more direct and tangible perspective of our method’s efficacy, we present the visual results of detection on selected scenes from the VisDrone dataset in Fig. 6. Evidently, YOLOv8 exhibits noticeable omissions in detecting extremely small objects, particularly in the distant background. The third scene, characterized by a multitude of object categories and instances, presents an intricate challenge exacerbated by occlusions. Notably, YOLOv8 struggles with the detection of pedestrians (indicated by the red bounding box). In contrast, our model noticeably outperforms YOLOv8, demonstrating superior detection capability even for small and occluded objects. This exemplifies our model’s remarkable aptitude for capturing intricate details and overcoming challenges posed by occlusions, leading to enhanced performance in complex scenes.

Table 5 Results on VisDrone datasets

Methods	mAP_{50}^{val}	$mAP_{50:95}^{val}$	mAP_{50}^{test}	$mAP_{50:95}^{test}$	Params (M)	GFLOPs
YOLOv5-n [19]	0.337	0.194	0.278	0.156	2.5	7.2
YOLOv5-s [19]	0.401	0.239	0.328	0.189	9.1	24.1
YOLOv5-m [19]	0.430	0.263	0.352	0.205	25.1	64.4
YOLOv6-n [20]	0.325	0.188	0.275	0.158	4.7	11.1
YOLOv6-s [20]	0.372	0.220	0.313	0.180	18.5	44.2
YOLOv6-m [20]	0.417	0.251	0.356	0.211	34.9	82.2
YOLOv7-tiny [21]	0.307	0.182	0.268	0.151	6.0	13.7
YOLOv8-n [22]	0.339	0.196	0.279	0.158	3.2	8.9
YOLOv8-s [22]	0.406	0.242	0.329	0.190	11.2	28.8
YOLOv8-m [22]	0.433	0.265	0.353	0.207	25.9	79.3
TA-YOLO-tiny	0.365	0.218	0.292	0.165	2.3	7.9
TA-YOLO-n	0.401	0.241	0.316	0.180	3.8	14.1
TA-YOLO-s	0.454	0.277	0.371	0.214	13.9	43.3
TA-YOLO-o	0.465	0.286	0.377	0.219	21.4	64.6
TA-YOLO-m	0.488	0.302	0.396	0.231	29.7	110.2

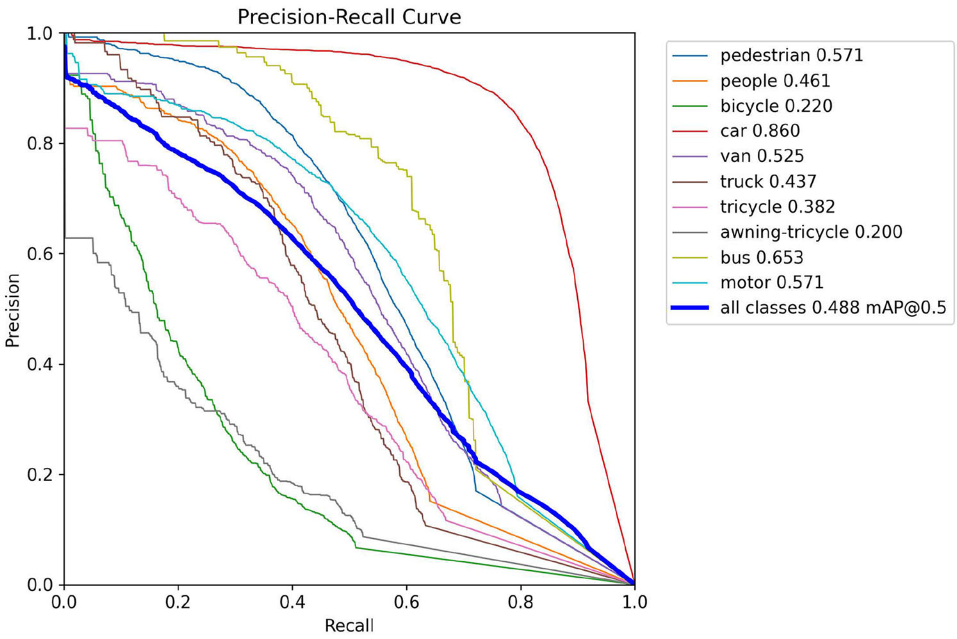

Fig. 5 Precision-recall curve on the VisDrone validation set

Table 6 Ablation experimental results on VisDrone validation set

Methods										
Baseline	P2 Head	Fusion	MCTA	MSTA	MCSTA	P	R	mAP^{50}	$mAP^{50:95}$	Params (M)
✓						0.450	0.338	0.339	0.196	3.2
✓	✓					0.468	0.368	0.375(↑3.6%)	0.224	3.4
✓	✓	✓				0.499	0.366	0.383(↑4.4%)	0.228	3.4
✓	✓	✓	✓			0.494	0.381	0.389(↑5.0%)	0.233	3.5
✓	✓	✓		✓		0.504	0.373	0.391(↑5.2%)	0.234	3.5
✓	✓	✓			✓	0.502	0.389	0.401(↑6.2%)	0.241	3.8

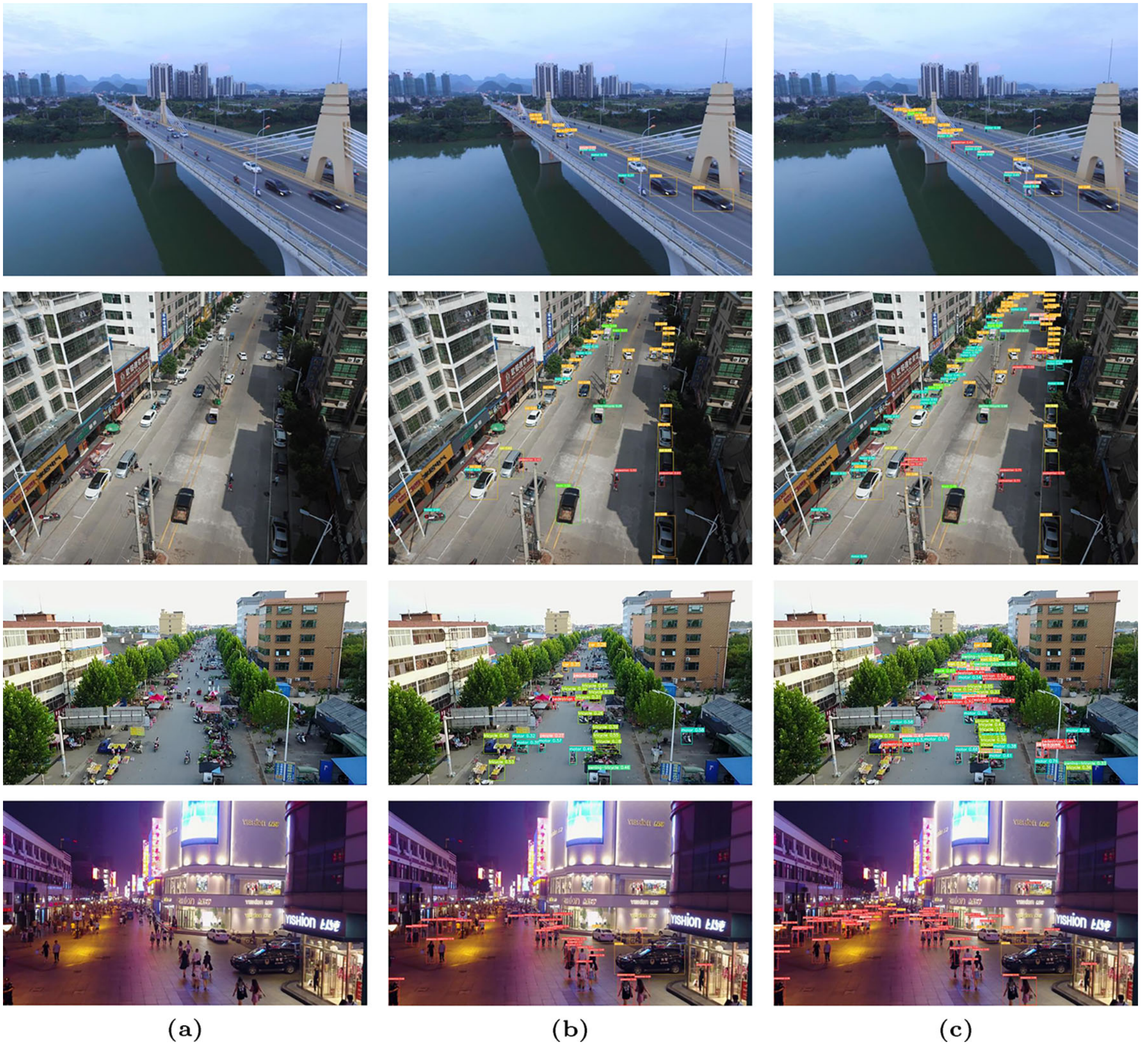

Fig. 6 The visualization results on VisDrone datasets. In **a** is the original image, in **b** is the result of YOLOv8-n, and in **c** is the result of the proposed TA-YOLO-n. Different object categories in the figure are boxed in different colors

Ablation experiments

To validate the effectiveness of our proposed approach in the context of small object detection, we conducted a series of ablation experiments on the VisDrone validation set, utilizing YOLOv8-n as the baseline model. The comprehensive results of these experiments are presented in Table 6. This set of ablation experiments highlights the incremental effect of each module or improvement we introduced, illustrating the diverse degree of performance enhancement achieved. The outcomes of these experiments unequivocally reaffirm the efficacy of our method in elevating network performance across the board.

Furthermore, to explore the impact of integrating the MCSTA module at different network stages, we conducted a series of experiments encompassing five configurations. Initially, we incorporated the module solely in the backbone, then sequentially added it to levels P2 through P5. The experimental results are shown in Table 7. By analyzing the outcomes of these experiments, a clear pattern emerged: the optimal effect is attained by incorporating the module at each stage. Remarkably, this module augmentation incurs a marginal increase in parameters.

Table 7 Ablation experimental results on VisDrone validation set

Backbone	P2	P3	P4	P5	mAP^{50}	$mAP^{50:95}$	Params (M)
✓					0.389	0.232	3.3
✓	✓				0.394	0.235	3.4
✓	✓	✓			0.395	0.237	3.4
✓	✓	✓	✓		0.398	0.238	3.5
✓	✓	✓	✓	✓	0.401	0.241	3.8

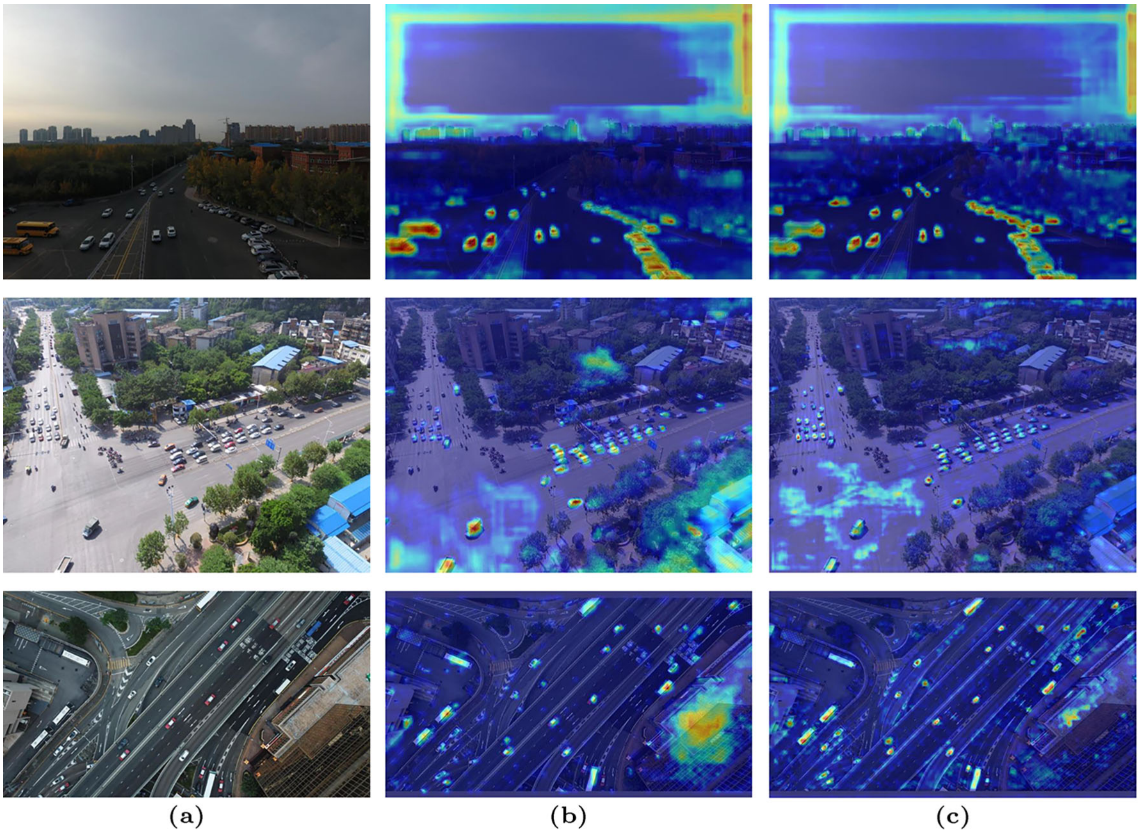

Fig. 7 Grad-CAM visualization. **a** is the original image, **b** and **c** are the visualization results of removing the MCSTA module and adding it, respectively

To provide a more visual understanding of the efficacy of the MCSTA module, we employed the gradient-based class activation map (Grad-CAM) method [53] to visualize the targeted network layer. Fig. 7 exhibits the feature visualization outcomes for the input detection head within the P3 level. The enhancement afforded by the MCSTA module becomes evident as the model captures a more comprehensive range of foreground information, thereby mitigating background interference while concurrently extracting finer-grained details. This visualization reaffirms the module's contribution to refining the model's feature extraction capabilities.

Conclusion

This paper proposes a novel multidimensional attention feature extraction module based on the Transformer architecture. This module not only accomplishes remote pixel interactions from channel and spatial dimensions but also integrates special linear convolutions, multi-head mechanisms, and intricately designed block attention patterns, enhancing the module's performance. These design choices not only enrich the module's feature representation but also significantly boost computational speed, ensuring high efficiency and practicality.This plug-and-play module exhibits strong versatility and can swiftly integrate into any natural image-based convolutional neural network for object detection. Its remarkable adaptability particularly shines in efficiently handling small objects within remote sensing images, empowering it with robust detection capabilities in the remote sensing domain. Moreover, owing to its lightweight nature, it's highly suitable for deployment on edge devices, significantly enhancing practicality and convenience for real-world applications.

Further experimental results validate the superiority of our model in object detection tasks, achieving superior performance with fewer parameters. Particularly in scenarios dealing with remote sensing images featuring small objects, our model demonstrates clear advantages. This not only provides a new technical solution for remote sensing image processing but also presents novel insights and possibilities for applications demanding efficient and precise detection of small-scale targets. In future research, we aim to apply this module to other fields requiring high-resolution pixel-level features, such as image segmentation and super-resolution domains. By applying this module in these areas, we aim to further explore and validate its feature capturing capabilities across diverse tasks and scenarios, laying a robust technical foundation and support for broader applications.

Data availability Data available on request from the authors.

Declarations

Conflict of 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.

References

1. Zou Z, Chen K, Shi Z, Guo Y, Ye J (2023) Object detection in 20 years: a survey. In: Proceedings of the IEEE
2. Li J, Xu R, Ma J, Zou Q, Ma J, Yu H (2023) Domain adaptive object detection for autonomous driving under foggy weather. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 612–622
3. Shen L, Tao H, Ni Y, Wang Y, Stojanovic V (2023) Improved yolov3 model with feature map cropping for multi-scale road object detection. Meas Sci Technol 34(4):045406
4. Mao J, Shi S, Wang X, Li H (2022) 3D object detection for autonomous driving: a review and new outlooks. arXiv:2206.09474
5. El-Ghamry A, Darwish A, Hassanien AE (2023) An optimized CNN-based intrusion detection system for reducing risks in smart farming. Internet Things 22:100709
6. Zhou W, Guan H, Li Z, Shao Z, Delavar MR (2023) Remote sensing image retrieval in the past decade: achievements, challenges, and future directions. In: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing
7. Liang Y, Han Y, Jiang F (2022) Deep learning-based small object detection: a survey. In: Proceedings of the 8th International Conference on Computing and Artificial Intelligence, pp 432–438
8. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88:303–338
9. Lin T-Y, Maire M, Belongie S , Hays J, Perona P, Ramanan D, Dollár P , Zitnick CL (2014) Microsoft coco: Common objects in context. In: Computer vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, Sept 6–12, 2014, Proceedings, Part V 13, pp 740–755. Springer
10. Wen L, Cheng Y, Fang Y, Li X (2023) A comprehensive survey of oriented object detection in remote sensing images. Expert Syst Appl 24:119960
11. Li C, Cheng G, Wang G, Zhou P, Han J (2023) Instance-aware distillation for efficient object detection in remote sensing images. IEEE Trans Geosci Remote Sens 61:1–11
12. Zhang J, Lei J, Xie W, Fang Z, Li Y, Qian D (2023) Superyolo: super resolution assisted object detection in multimodal remote sensing imagery. IEEE Trans Geosci Remote Sens 61:1–15
13. Gao L, Liu B, Ping F, Mingzhu X (2023) Adaptive spatial tokenization transformer for salient object detection in optical remote sensing images. IEEE Trans Geosci Remote Sens 61:1–15
14. Liu Y, Yuan Y, Wang Q (2023) Uncertainty-aware graph reasoning with global collaborative learning for remote sensing salient object detection. In: IEEE Geoscience and Remote Sensing Letters
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
16. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
17. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv:1804.02767
18. Bochkovskiy A, Wang C-Y, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. arXiv:2004.10934
19. Jocher G (2022) Yolov5. code repository https://www.github.com/ultralytics/yolov5
20. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, Ke Z, Li Q, Cheng M, Nie W et al (2022) Yolov6: a single-stage object detection framework for industrial applications. arXiv:2209.02976
21. Wang C-Y, Bochkovskiy A, Liao HYM (2023) 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, pp 7464–7475
22. Jocher G (2023) Yolov8. code repository https://github.com/ultralytics/ultralytics
23. Wang K, Liew JH, Zou Y, Zhou D, Feng J (2019) Panet: few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9197–9206
24. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: 2017 The Thirty-first Conference on neural information processing systems (NeurIPS), pp 5998–6008
25. Du D, Zhu P, Wen L, Bian X, Lin H, Hu Q, Peng T, Zheng J, Wang X, Zhang Y , et al (2019) Visdrone-det2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF international conference on computer vision workshops
26. 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
27. Girshick Ross (2015) Fast r-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
28. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: 2015 The Twenty-nine Conference on neural information processing systems (NeurIPS), pp 91–99
29. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
30. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
31. Sandler M, Howard A , Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In :Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
32. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for

mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324

33. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6848–6856
34. Ma N, Zhang X, Zheng H-T, Sun J (2018) Shufflenet v2: practical guidelines for efficient CNN architecture design. In: Proceedings of the European conference on computer vision (ECCV), pp 116–131
35. Liu W , Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (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
36. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
37. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning (PMLR), pp 6105–6114
38. Hu P, Ramanan D (2017) Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 951–959
39. Zheng Z, Zhong Y, Ma A, Han X, Zhao J, Liu Y, Zhang L (2020) Hynet: hyper-scale object detection network framework for multiple spatial resolution remote sensing imagery. ISPRS J Photogram Remote Sens 166:1–14
40. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S , Fu C-Y, Berg AC (2016) Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, Oct 11–14, 2016, Proceedings, Part I 14, pp 21–37. Springer
41. Fu C-Y, Liu W, Ranga A, Tyagi A, Berg AC (2017) DSSD: Deconvolutional single shot detector. arXiv:1701.06659
42. Xiang W, Zhang D-Q, Yu H, Athitsos V (2018) Context-aware single-shot detector. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1784–1793
43. Cao G, Xie X, Yang W, Liao Q , Shi G, Wu J (2018) Feature-fused SSD: fast detection for small objects. In: Ninth international conference on graphic and image processing (ICGIP 2017), vol 10615, pp 381–388
44. Bell S, Zitnick CL, Bala K, Girshick R (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2874–2883
45. Bai Y, Zhang Y, Ding M, Ghanem B (2018) SOD-MTGAN: Small object detection via multi-task generative adversarial network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 206–221
46. Noh J, Bae W, Lee W, Seo J, Kim G (2019) Better to follow, follow to be better: towards precise supervision of feature super-resolution for small object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9725–9734
47. Mnih V, Heess N, Graves A et al (2014) Recurrent models of visual attention. In: 2014 The Twenty-nine Conference on neural information processing systems (NeurIPS), pp 2204–2212
48. 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
49. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
50. Hou Q, Zhou D, Feng J (2021) Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13713–13722
51. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1580–1589
52. Liu Z, Hu H, Lin Y, Yao Z, Xie Z, Wei Y, Ning J , Cao Y, Zhang Z, Dong L, et al (2022) Swin transformer v2: scaling up capacity and resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12009–12019
53. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.