Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

SRE-Ret: An Object Detection Method Based on Sparse Region Extraction

Yanming Ye[1,2*], Kailong Cheng[1], Qiang Sun[3], Xingfa Shen[1], Xinyi Chang[4]

[1] School of Computer Science, Hangzhou Dianzi University, Hangzhou, 310018, China
[2] School of Information Engineering, Hangzhou Dianzi University, Hangzhou, 311305, China
[3] School of Science, Zhejiang Sci-Tech University, Hangzhou, 310018, China
[4] School of Data Science and Intelligent Media, Communication University of China, Beijing, 100024, China
E-mail: yeym@hdu.edu.cn

Received: 13 May 2025; **Revised:** 17 July 2025; **Accepted:** 26 August 2025

Abstract: The continuous advancement of image acquisition technology and the subsequent proliferation of high-resolution images have introduced significant challenges to conventional object detection methodologies. While high-resolution feature maps offer a distinct advantage in detecting small objects due to their retention of detailed information, the concomitant increase in candidate regions and computational complexity substantially impedes real-time performance. Conversely, low-resolution feature maps, although computationally efficient, often lack the necessary precision for effective small object detection, failing to satisfy practical application demands. Consequently, optimizing the allocation of computational resources within high-resolution feature maps while preserving the accuracy of small object detection has emerged as a critical focus and ongoing challenge in contemporary research. To address these limitations, this paper introduces an object detection method based on Sparse Region Extraction (SRE), termed SRE-Ret. This method leverages the window-based and shifted-window self-attention mechanisms inherent in the Swin-Transformer architecture. By employing sparse region selection on high-resolution feature layers, it selectively filters feature windows likely to contain objects, thereby substantially reducing the number of candidate regions and redundant computations. Furthermore, a dedicated small object detection head is integrated into the high-resolution feature layers for precise prediction, while an efficient convolutional detection head is utilized on the low-resolution feature layers for rapid inference. The novelty of this approach lies in achieving sparse processing of feature regions via the SRE module, effectively balancing precision and efficiency in multi-scale feature detection.

***Keywords*:** object detection, transformer, attention mechanism, sparse region extraction, small object detection, embedded devices

MSC: 68T01, 68T07, 68T20

1. Introduction

Object detection, a fundamental technology in the field of computer vision, seeks to identify instances of specific object categories within images or videos and determine their spatial locations. Driven by the rapid development of artificial intelligence, object detection has found widespread application across a multitude of fields.

DOI: https://doi.org/10.37256/cm.6520257227

The advent of deep learning, and in particular the proliferation of Convolutional Neural Networks (CNNs), has led to a significant advancement in object detection performance. CNN-based object detection methodologies can be broadly classified into two primary types: two-stage detection and single-stage detection, distinguished by their underlying detection processes.

The genesis of two-stage object detection methods can be attributed to Girshick et al.'s Region-based Convolutional Neural Network (R-CNN) [1], introduced in 2014. R-CNN pioneered the application of CNNs to object detection, employing Selective Search to generate approximately 2,000 candidate regions and extracting features from each for classification and localization. In 2015, Girshick [2] presented Fast R-CNN, which incorporated Region of Interest (RoI) Pooling to enable shared feature extraction across all candidate regions, thereby significantly reducing computational overhead and optimizing training and inference. Ren et al. [3] further advanced the field with Faster R-CNN, integrating candidate region generation into the network via the Region Proposal Network (RPN). In 2017, He et al. [4] introduced Mask R-CNN, an extension of Faster R-CNN that incorporated a mask branch to enable instance segmentation, achieving pixel-level object delineation. This approach allows for precise object contour mapping, making it applicable to both semantic and instance segmentation tasks. Cascade R-CNN, proposed by Cai and Vasconcelos [5] in 2018, utilizes a cascaded architecture with multiple detection heads to progressively refine the detection of high-quality targets, effectively reducing false positives. Furthermore, Lin et al. [6] introduced the Feature Pyramid Network (FPN), which constructs a feature pyramid to integrate multi-scale feature information, substantially improving the model's ability to detect objects across a range of scales, particularly small objects.

The foundational work in single-stage detection methods was presented in 2016 by Redmon et al. [7] with You Only Look Once (YOLO). YOLO reframes object detection as a regression problem by partitioning the image into a grid and directly predicting object classes and bounding box coordinates within each grid cell. This approach facilitates end-to-end real-time detection, establishing a new paradigm in single-stage detection. Subsequent YOLO iterations have continued to advance the field: YOLOv2-YOLOv4 improved YOLO in multi-scale training, multi-scale prediction, and the introduction of CIoU loss and DIou Non-Maximum Suppression (NMS) respectively [8–10].YOLOv5 offers multiple pre-trained models and configuration options, suitable for different application scenarios and hardware environments. Its modular design and PyTorch implementation have improved flexibility and scalability [11]. YOLOv6 further reduces the number of model parameters and computational complexity while maintaining high performance, making it suitable for running on resource constrained devices. At the same time, the feature extraction module has been optimized, improving the detection capability for small targets [12]. YOLOv7 introduces more training techniques, such as adaptive learning rate adjustment and dynamic label smoothing, to improve the training performance of the model. At the same time, more advanced data augmentation techniques were used to improve the robustness and generalization ability of the model [13]. YOLOv8 has achieved a leap from fast detection to versatile visual tools in the YOLO series through Anchor Free architecture, Path Aggregation Feature Pyramid Network (PAFPN) feature fusion, multi task unified framework, and deployment optimization [14].

Furthermore, in 2017, Facebook AI Research (FAIR) [15] introduced RetinaNet, which significantly enhanced the performance of single-stage detectors through an innovative loss function and a multi-scale feature processing mechanism. This enabled RetinaNet to achieve detection accuracy comparable to two-stage detectors while maintaining its speed advantage.

The success of Transformer models in natural language processing (e.g., Bidirectional Encoder Representations from Transformers (BERT) [16] and Generative Pre-Training (GPT) [17]) has spurred computer vision researchers to adapt them for visual tasks. The core of the Transformer architecture lies in its attention mechanism, which addresses CNN's limitations in capturing long-range dependencies by enabling global feature modeling.

In 2020, Detection Transformer (DETR) [18], introduced by Facebook AI Research, pioneered the application of Transformers to object detection, establishing an end-to-end detection paradigm. DETR eliminates conventional post-processing steps such as candidate box generation and NMS, leveraging the Transformer's encoder and decoder to directly predict the set of objects, thereby simplifying the detection pipeline.

Despite its innovative design, DETR exhibits certain limitations. First, DETR's training efficiency is suboptimal, characterized by slow model convergence, often necessitating hundreds of epochs to achieve peak performance,

significantly exceeding the training requirements of traditional CNN models. Second, DETR exhibits challenges in detecting small objects due to the high computational complexity of the Transformer's attention mechanism when applied to high-resolution feature maps, which impedes the extraction of fine-grained features. Furthermore, DETR's computational complexity scales quadratically with input image resolution, thus restricting its applicability to high-resolution imagery.

To address DETR's shortcomings, researchers have proposed several enhancements. For instance, Deformable DETR [19] introduces a deformable attention mechanism, reducing computational complexity by focusing on sparse key feature points, while simultaneously enhancing convergence speed and detection accuracy. Efficient DETR [20] optimizes the Transformer architecture and training strategy to further improve model efficiency. While these advancements mitigate some of DETR's limitations, significant optimization potential remains, particularly in small object detection and high-resolution image processing. Liu et al. [21] introduced the Swin Transformer, which incorporates a hierarchical structure and a shifted window attention mechanism to effectively capture multi-scale features. Similarly, Wang et al. [22] proposed the Pyramid Vision Transformer model in 2021, enhancing the model's adaptability to objects of varying scales through pyramidal feature extraction. Zhang et al. [23] advanced the DETR framework by proposing the DETR with Improved Denoising anchor boxes (DINO) model, which improves performance through an enhanced denoising strategy. DINO introduces denoising anchor boxes, dynamically adjusting query vectors during training to enable the model to better focus on target regions. Byungseok et al. [24] proposed Sparse DETR that selectively updates only the tokens expected to be referenced by the decoder, thus help the model effectively detect objects.

Beyond purely Transformer-based models, researchers have explored hybrid approaches that integrate Transformers with Convolutional Neural Networks (CNNs) to leverage the complementary strengths of both architectures. Srinivas et al. [25] proposed BoTNet in 2021, integrating an attention mechanism into the classic ResNet architecture by replacing the final convolutional layer with a Transformer module. This design retains the efficiency of CNNs in local feature extraction while significantly enhancing the model's capacity to model global features. Peng et al. [26] introduced the Conformer in 2021, which employs an innovative dual-branch structure. One branch utilizes CNNs to focus on extracting local detailed features, while the other leverages Transformers to capture global contextual information. The features from both branches are fused at a later stage, effectively improving detection accuracy. Dai et al. [27] proposed CoAtNet, which achieves a balance between performance and computational efficiency by deeply integrating convolutional and attention mechanisms. Sachin et al. [28] proposed MobileViT, which reduces computational complexity by reducing the number of attention heads, using smaller hidden dimensions, and introducing low rank approximations, ensuring that the model can run efficiently on mobile devices.

Small object detection remains a persistent challenge, particularly in high-resolution images where small objects occupy a limited pixel proportion, exhibit sparse feature information, and are susceptible to being overwhelmed by background noise. Traditional methods often suffer from missed or false detections due to insufficient feature map resolution or inaccurate candidate box generation. To address this, recent targeted approaches include the Feature Pyramid Network (FPN) [6], which enhances small object representation through multi-scale feature fusion. Scale Normalization for Image Pyramids with Efficient Resampling (SNIPER) [29] and QueryDet [30], which improve small object detection via multi-scale training and sparse query strategies. However, these methods still face limitations in computational efficiency and real-time performance, especially in resource-constrained environments.

Currently, object detection technology faces significant challenges, including training efficiency, computational complexity, and the accuracy of small object detection. Specifically, anchor-based approaches exhibit a geometric increase in the number of candidate boxes when processing high-resolution imagery, thereby significantly amplifying computational demands. The persistence of these challenges hinders the continued advancement of object detection technology in practical applications.

To address the issue of a large number of candidate boxes and the associated computational burden in high-resolution images, we propose a small object detection framework based on Sparse Region Extraction (SRE). This method intelligently selects local regions that are likely to contain targets through the SRE module, significantly reducing redundant computations and thereby improving detection efficiency. To further enhance the detection capability for small objects, we designed a dedicated small object detection head to optimize the feature representation of small-scale targets.

Experimental results on the VisDrone dataset demonstrate that this method performs excellently in complex scenes such as high-resolution images captured by drones, with small object detection accuracy significantly surpassing that of baseline methods, validating its efficiency and practicality.

2. Relation work

2.1 *Retinanet*

RetinaNet [15], introduced by Facebook AI Research (FAIR) in 2017, is a single-stage object detection algorithm specifically designed to address the accuracy limitations of traditional single-stage detectors in complex scenes and for small object detection. While single-stage detectors are favored for their efficiency compared to two-stage detectors, they often exhibit lower accuracy. RetinaNet bridges this gap by introducing an innovative Focal Loss function and a multi-scale feature processing mechanism, significantly enhancing the performance of single-stage detectors. It achieves detection accuracy comparable to two-stage detectors while retaining the speed advantage, with particularly notable performance on large-scale datasets like Common Objects in Context (COCO).

Figure 1 illustrates the overall architecture of RetinaNet. The architecture comprises several key components. First, a backbone network, typically ResNet (e.g., ResNet-50 or ResNet-101), extracts deep features from the input image. These features are then fed into a Feature Pyramid Network (FPN), which fuses features across different scales via a top-down pathway and lateral connections to generate multi-scale feature maps. These feature maps, denoted P3 to P7, cater to different object sizes: P3 provides high-resolution features suitable for small objects, while P7 provides low-resolution features ideal for large objects. This multi-scale design enables RetinaNet to effectively handle the diversity of object sizes in images, thereby improving detection robustness. Subsequently, classification and regression subnets perform class prediction and bounding box regression on anchor boxes across these feature maps. Anchor boxes are predefined sets of boxes with varying scales and aspect ratios, designed to cover potential object regions and are aligned with the feature map levels to ensure accurate detection.

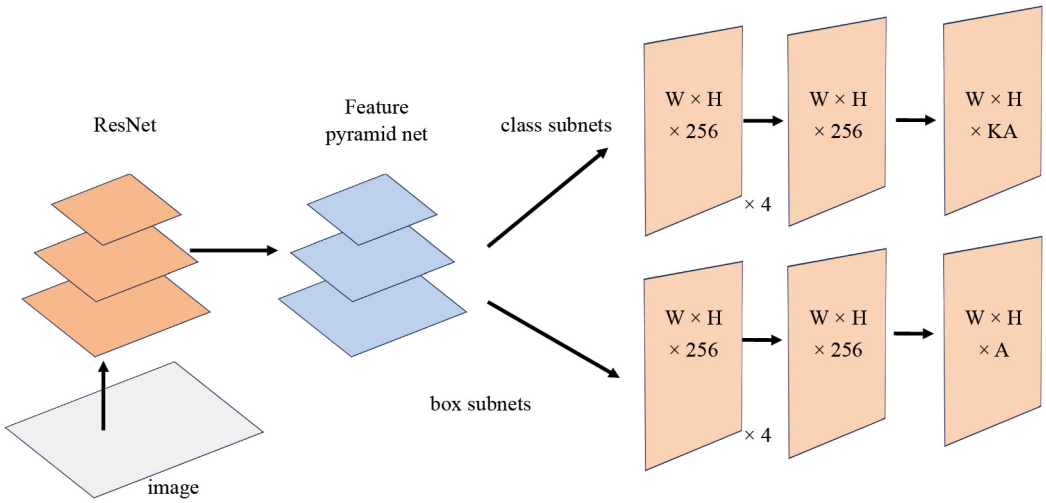

Figure 1. Retinanet architecture

The most significant innovation of RetinaNet is the Focal Loss, a dynamically weighted cross-entropy loss designed to address the class imbalance problem prevalent in object detection. In such tasks, background samples (easy to classify) vastly outnumber foreground object samples (hard to classify), leading models to focus on the abundant background samples while neglecting challenging foreground object samples. Focal Loss dynamically adjusts sample weights to

emphasize hard-to-classify samples, thereby improving detection accuracy. This design effectively shifts the model's focus to difficult samples, significantly enhancing detection capabilities for small and occluded objects.

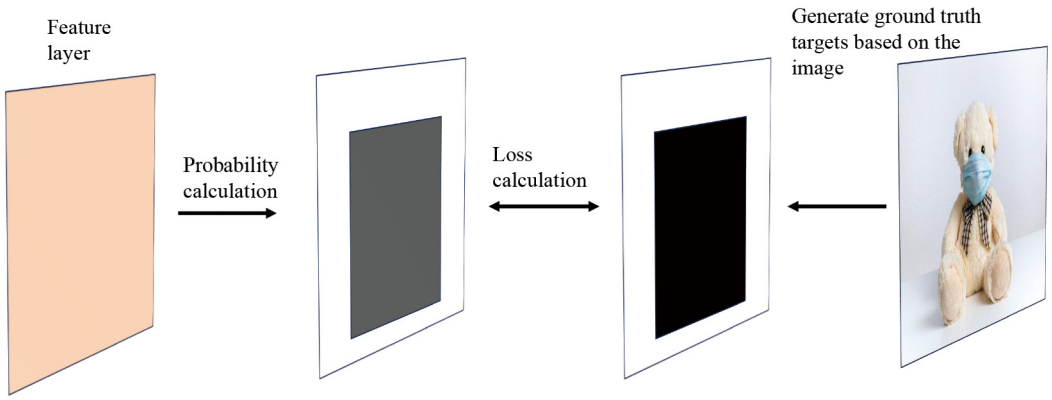

Figure 2. Query Loss Process Illustration

For the loss function design, RetinaNet employs Focal Loss for classification and Smooth L1 Loss for bounding box regression. The total loss is a weighted sum of both, with hyperparameters tuning their relative importance to optimize both tasks synergistically. RetinaNet's strength lies in its combination of efficiency and high accuracy. Traditional single-stage detectors struggle with complex scenes due to the lack of a region proposal stage, but RetinaNet mitigates the class imbalance issue through Focal Loss, achieving accuracy comparable to two-stage detectors while preserving the high speed of single-stage methods. Experiments on the COCO dataset demonstrate that RetinaNet performs comparably to two-stage methods like Faster R-CNN, with a clear advantage in inference speed. Additionally, its simple and extensible design has made it a seminal model in single-stage object detection.

In summary, RetinaNet significantly advances single-stage object detection through its innovative use of Focal Loss and a multi-scale feature pyramid. Its exceptional performance in handling complex scenes and small objects not only drives progress in object detection technology but also provides valuable insights for future research.

2.2 *Swin-transformer*

In the realm of Transformer-based object detection methodologies, DETR [18] marked a seminal advancement by introducing Transformers into object detection through the adoption of set prediction and bipartite matching. This innovative approach eliminates the reliance on conventional anchor boxes and Non-Maximum Suppression (NMS) procedures, thereby streamlining the detection pipeline and facilitating end-to-end optimization. In contrast, the Swin Transformer [21] optimizes the computational complexity inherent in Transformer applications within image processing by incorporating a hierarchical design and a shifted window mechanism, while preserving robust feature extraction capabilities.

The Swin Transformer is engineered with the objectives of reducing computational complexity, enhancing local feature modeling, and maintaining a degree of global information interaction. Its architecture and operational mechanisms can be distilled into several key aspects:

Hierarchical Feature Extraction: Unlike the traditional Vision Transformer (ViT) [31], which segments images into fixed-size patches for global modeling, the Swin Transformer employs a hierarchical patch merging strategy. Initially, the input image is divided into small patches (e.g., 4 × 4 pixels), which are subsequently merged across multiple stages. This process reduces the spatial resolution of feature maps while increasing the dimensionality of the channel. Analogous to pooling operations in Convolutional Neural Networks (CNNs), this design enables the capture of multiscale features across different levels, decreasing the number of tokens and improving adaptability to objects of varying scales.

Shifted Window Self-Attention: The Swin Transformer's primary innovation lies in its localized self-attention mechanism, termed Shifted Window Self-Attention. While traditional global self-attention incurs a computational complexity of $O(N^2)$, the Swin Transformer partitions the feature map into non-overlapping local windows, computing self-attention exclusively within each window. This reduces complexity to $O(W^2)$, where W represents the number of tokens within a window. Such a localized approach significantly lowers computational overhead while enhancing perception of local features, akin to convolutional operations in CNNs.

Shifted-Window Strategy: To mitigate information isolation between local windows, the Swin Transformer introduces a shifted window strategy. In consecutive Transformer layers, window positions alternate, enabling tokens in adjacent windows to interact via the attention mechanism. This design ensures efficient local computation while indirectly facilitating global information flow, striking a balance between efficiency and modeling capacity.

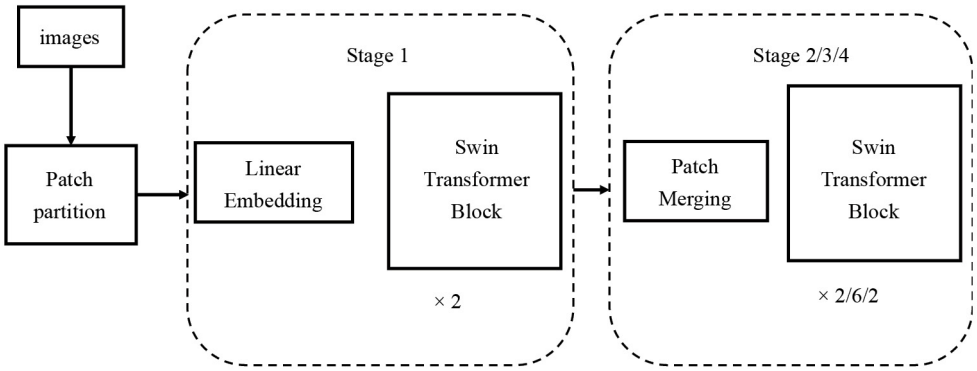

Figure 3. Swin Transformer Architecture

The Swin Transformer comprises multiple stages, this is shown in Figure 3, each encompassing a patch merging layer and several Transformer blocks. The operational sequence is as follows:

Patch Partition: The input image is segmented into initial patches.

Linear Embedding: These patches undergo linear projection to generate initial feature representations.

Stage 1: Local window self-attention is applied to high-resolution feature maps, extracting fine-grained features.

Patch Merging and Subsequent Stages: Patches are merged, and self-attention computations are repeated to progressively derive higher-level features.

This hierarchical framework enables efficient processing of image data across varying resolutions.

The self-attention mechanism in the Swin Transformer is computed independently within each window, as delineated by the following equation (1):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}} + B\right)V \tag{1}$$

Here Q, K, and V denote the query, key, and value matrices, respectively. d represents the feature dimension, and B signifies the relative position bias, which encodes spatial relationships among tokens within the window to bolster structural awareness. The shifted window strategy alternates between regular and offset windows across layers, ensuring cross-window information exchange.

In terms of computational complexity, the Swin Transformer's self-attention mechanism achieves a complexity of $O(2M^2HW)$, where M is the window size, and H and W are the height and width of the feature map, respectively. This is markedly lower than the $O((HW)^2)$ complexity of global self-attention. The shifted window mechanism further sustains global modeling capabilities, rendering the design computationally efficient yet expressive.

Within object detection tasks, the Swin Transformer typically functions as a backbone network, supplanting traditional CNN-based architectures. Its hierarchical feature maps integrate seamlessly with Feature Pyramid Networks (FPN), supporting multi-scale object detection. Unlike DETR's global feature processing, the Swin Transformer's localized and hierarchical design excels in efficiently handling high-resolution images, particularly in detecting small objects. When evaluated on the COCO dataset and paired with detection heads such as Faster R-CNN or Mask R-CNN, the Swin Transformer markedly enhances detection accuracy, especially in complex scenes and small object detection scenarios.

3. Method

This method aims to optimize computational efficiency and detection accuracy in object detection tasks through a hierarchical processing strategy. Its core idea is to divide the multi-scale features of the input image into high-resolution feature layers and low-resolution feature layers, and to adopt different processing methods for features of different resolutions. Specifically, for high-resolution feature layers, the Sparse Region Extraction (SRE) module is used to select feature windows that may contain targets, reducing the number of candidate regions, and predictions are made through the small object detection head module. For low-resolution feature layers, the convolutional detection head module is directly used for efficient global detection. Figure 4 shows the overall architecture of this model.

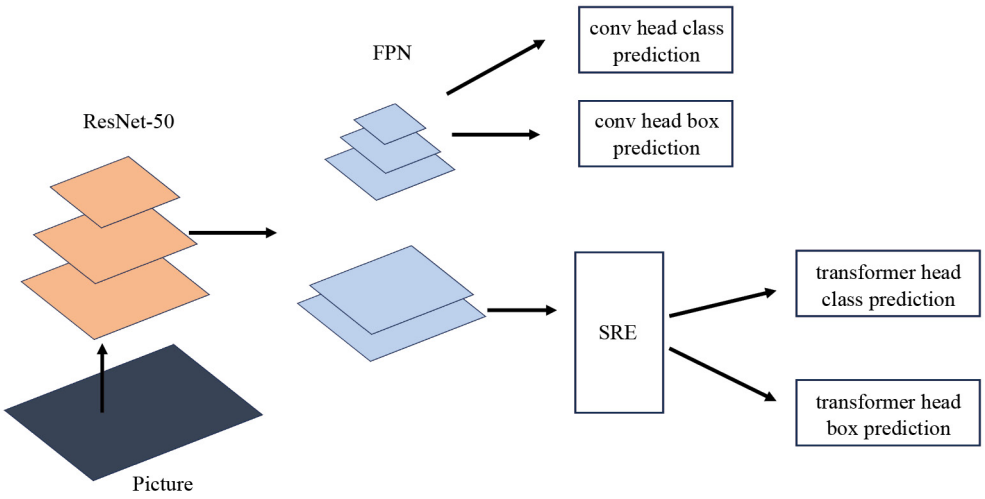

Figure 4. Overall Framework of the SRE-Ret Model

3.1 *Sparse region extractor*

The SRE module (Sparse Region Extractor) is a core component in this object detection model, primarily designed to process high-resolution feature maps to select key regions that may contain small targets. The design of this module aims to reduce the computational complexity of subsequent detection while ensuring the detection accuracy of small targets. It takes the high-resolution enhanced features from the feature fusion module as input and achieves feature compression and extraction of key regions through a series of processing steps.

The SRE module first performs feature extraction on the output of the FPN. The data processing flow in this part includes several key steps. First, the input feature map is divided into multiple small patches through the patch embedding layer. Since we are only concerned with whether there are targets in the region, the feature map is mapped to low-dimensional vectors. Then, these feature vectors are processed through the window multi-head self-attention layer and the shifted window multi-head self-attention layer, using the window mechanism to capture relationships between local

features while introducing global contextual information. Subsequently, the layer normalization layer standardizes the features to ensure numerical stability. Finally, the features are compressed into a single-channel feature map through a convolutional layer for subsequent region selection. This part can be formally defined as: $F_{saliency} = Conv_{3\times3}(F) \in \mathbb{R}^{H\times W\times 1}$ where $F \in \mathbb{R}^{H\times W\times C}$ is an input feature map and the convolution is bias-free to avoid introducing spurious activations. Here, a 3 × 3 convolution compresses the channel dimension to 1, producing a single-channel saliency map.

Figure 5 illustrates this process.

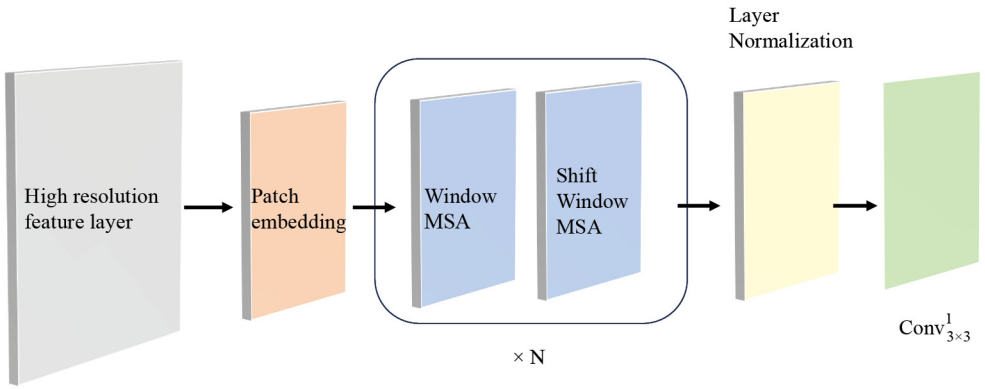

Figure 5. Structural Diagram of the SRE Module

In the selection phase, we use a sliding window approach to compute the single-channel feature map output from the window self-attention feature extraction. In each window, we calculate its probability value. Based on these probability values, the module selects a portion of the windows most likely to contain small targets as output. These selected feature windows provide high-quality input for subsequent detection, not only reducing redundant computations but also enhancing the model's focus on small targets.

The window selection operation is a crucial step in the SRE module, aiming to filter out regions from the single-channel feature map that are most likely to contain small targets. This operation quantifies the likelihood of each window containing a target, sorts them in descending order, and selects the top portion of regions for subsequent prediction, achieving sparse region extraction.

Specifically, the SRE module first divides the single-channel feature map into multiple fixed-size windows (e.g., 8 × 8). This part can be formally defined as: $\{M_i\}_{i=1}^{N}$. Here, The saliency map is divided into N non-overlapping windows of size $k \times k$. In our experiments, $k = 8$.

For each window, it calculates the maximum and average values of the pixels within it and obtains a probability value through a weighted sum, which serves as the basis for selecting the window. The calculation formula is typically a linear combination of the maximum and average values, given by Equation (2):

$$score_i = \alpha * max(Win_i) + (1 - \alpha) * mean(Win_i) \tag{2}$$

where $max(Win_i)$ represents the maximum feature value within the window, $mean(Win_i)$ represents the average feature value within the window, and α is learned weighting parameter (initialized at 0.5).

Subsequently, the module sorts the probability values of all windows in descending order and selects the top M windows with the highest probabilities as feature windows. M can be a fixed value or selected based on a certain proportion of the total number of windows. These feature windows are considered the key regions most likely to contain small targets, and subsequent detection is performed only within these regions, thereby avoiding computations in areas without target

objects and reducing redundant calculations. This part can be formally defined as: $W_{selected} = \{W_i \mid score_i$ ranks in the top m scores$\}$. where, $m \ll N$, and this reduces computation from $O(HWC)$ to $O(m^2C)$.

Based on the M feature windows selected in the previous step, their specific positions in the original high-resolution feature map are determined. Local feature blocks corresponding to each feature window are extracted from the high-resolution feature map. Assuming the original feature map has C channels, each local feature block has a size of [C, 8, 8], corresponding to the 8 × 8 size of the window.

After extraction, M local feature blocks are obtained, each retaining the complete information of the original feature map in the corresponding region. Then, the scattered local feature blocks are integrated into a unified feature map for subsequent processing. The M local feature blocks (each of size [C, 8, 8]) are concatenated along the height dimension H. The size of the concatenated feature map becomes [C, M × 8, 8], with the number of channels C remaining unchanged, and the height becomes M × 8 because the M blocks are stacked along the height direction, and the width remains 8, consistent with the width of a single feature block. The concatenated feature map is then input into the object detection head module for predicting the category and location of targets. Figure 6 provides a schematic diagram of this process.

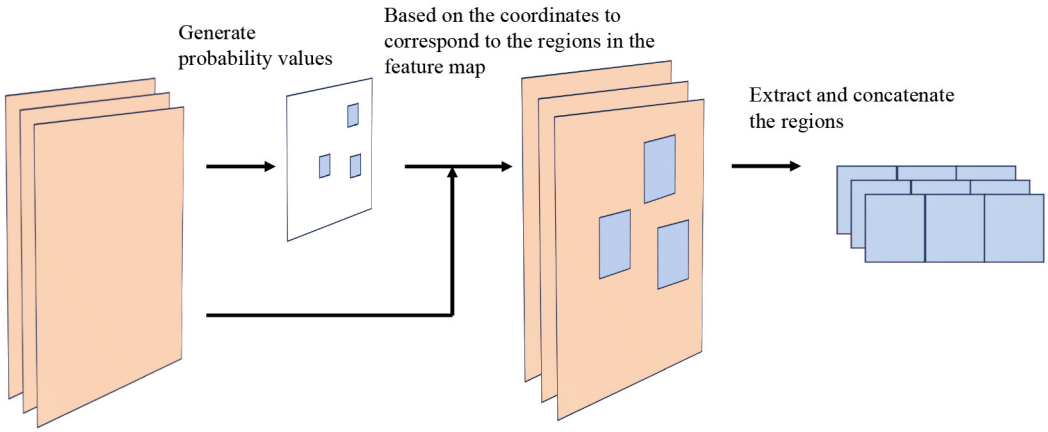

Figure 6. Feature Extraction and Concatenation Process

The advantage of the SRE operation lies in its focus on high-probability regions, significantly reducing the processing scope of high-resolution feature maps, thereby lowering computational costs. At the same time, it retains information relevant to small targets, ensuring that detection accuracy is not compromised. This probability-based window selection strategy is a key innovation of this model for efficiently detecting small targets.

3.2 *Detection head*

The detection head module is a critical component in the object detection pipeline, responsible for target classification and localization. In this system, two types of detection head modules are designed, each optimized for feature maps of different resolutions, balancing the detection needs of small targets and global targets.

The first detection head module focuses on detecting small-sized targets, processing the high-resolution feature windows selected by the SRE module. It includes two branches: the classification prediction branch and the bounding box prediction branch. These two branches consist of a combination of N window multi-head self-attention layers, layer normalization layers, and convolutional layers, respectively predicting the class and position offset of each prior box. Due to the sparsification of the input features through the SRE module, the first detection head can efficiently focus on small target regions, enhancing detection precision. Figure 7 illustrates the structure of this module.

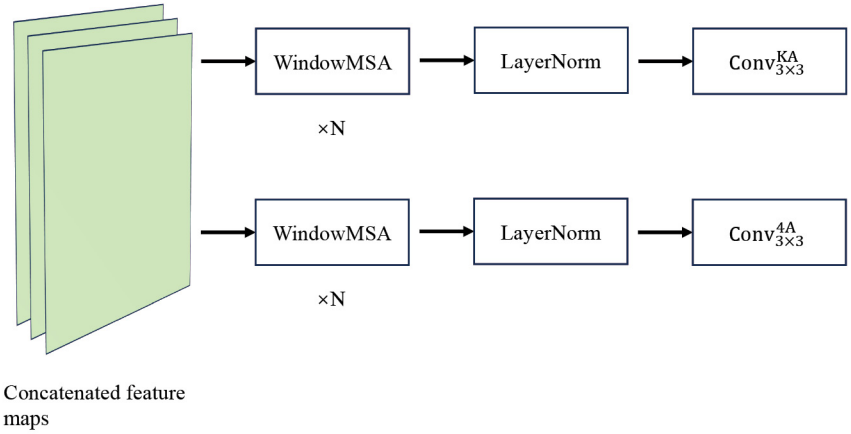

Figure 7. Structure of the Transformer-based Detection Head Module

The second detection head module is a conventional convolutional detection head, designed for global target detection on low-resolution feature maps. It also contains a classification prediction branch and a box prediction branch, predicting the class and position information of prior boxes through a multi-layer convolutional structure. Unlike the first detection head, the second detection head prioritizes computational efficiency and is suitable for detecting medium and large targets, with input directly from the low-resolution features output by the feature fusion module.

The collaborative operation of these two detection heads achieves a balance in the model's performance across targets of different scales. The transformer-based detection head accurately captures small targets, while the convolutional detection head covers global targets, thereby enhancing overall detection performance.

The loss function of this model is composed of the following three components:

Query Loss, which measures the discrepancy between the single-channel feature map output by the SRE module and the ground truth annotated feature map, can be regarded as a binary classification problem. The calculation formula is given by Equation (3):

$$loss_{query} = Smooth_{L1}(Sigmoid(SRE(feature)), GT) \tag{3}$$

Here, $SRE(feature)$ denotes the output feature map of the SRE module, i.e., the probability map. GT represents the ground truth annotated feature map, where regions with objects are labeled as 1 and the rest as 0. $SmoothL_1$ is the smooth L1 loss function, whose formula is given by Equation (4). This operation is illustrated in Figure 2.

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{if } |x| \geq 1 \end{cases} \tag{4}$$

Classification Loss, which evaluates the discrepancy between the predicted class and the true class of the detection boxes. It is calculated using the Focal Loss with. Its mathematical formulation is given by Equation (5),

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{5}$$

where p_t is the predicted probability for the true class, α_t is a balancing factor (e.g., 0.25) to reduce the influence of negative samples, and γ (typically 2) is a modulating factor that further decreases the weight of easy-to-classify samples.

Bounding Box Loss, which measures the error between the predicted offset and the true offset of the detection boxes. It is computed using the L1 norm loss function with formula (6):

$$Loss_{L1} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

where, y_i is the true value of the i-th sample, $hat y_i$ is the predicted value of the i-th sample, and n is the total number of samples.

The Total Loss is the weighted sum of the three aforementioned loss components, with the calculation formula given by Equation (7):

$$Loss = \lambda_1 1oss_{query} + \lambda_2 loss_{cls} + \lambda_3 loss_{box} \tag{7}$$

Here, λ_1, λ_2, λ_3 are weighting hyperparameters used to balance the contributions of each loss component.

3.3 *Object detection pipeline*

To streamline deployment and usage, we encapsulate the core algorithm into a unified Object Detection Pipeline. Users only need to initialize the pipeline and call its methods to perform object detection without handling low-level details. The main contents of the pipline encapsulation in this article include:

1) Multi-scale Anchor Generation

• First-stage features (P1-P2): Generate K_1 anchors per spatial position using aspect ratios $R_1 = r_{11}, ..., r_{1n}$ and scales $S_1 = s_{11}, ..., s_{1m}$.

• Second-stage features (P3-P6): Generate K_2 anchors per position with R_2 and S_2 following pyramidal scaling

• Total anchors $N = \sum_1^2 (H_i * W_i * K_i)$.

2) Dual-branch Prediction

For each feature level:

• Classification branch: f_{cls}: $\mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times K\times |C|}$ via 5-layer 3 × 3 CNN.

• Regression branch: f_{reg}: $\mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times K\times 4}$ via identical architecture.

• Output activation: Softmax for classification, linear for regression.

3) Box Refinement

For each anchor $a = (x_a, y_a, w_a, h_a)$ with predictions (c, δ):

• Refined coordinates: $b = (x_a + w_a \cdot \delta_x, y_a + h_a \cdot \delta_y, w_a \cdot e^{\delta_w}, h_a \cdot e^{\delta_h})$.

• Confidence score: s = max($\sigma(c)$) where σ is softmax.

4) Adaptive NMS

Input: All refined boxes B = $\{b_i\}$ with scores S = $\{s_i\}$

• Sort B by S in descending order.

• Initialize detection set D = $\emptyset$.

• While $B \neq \emptyset$, Select $b* = argmax(S)$; $DD \cup b*$; Remove all $b \notin B$ when $IoU(b, b) > \theta(\kappa(b))$, where $\theta(\cdot)$ is size-dependent threshold function and $\kappa(\cdot)$ returns box size category.

5) Output Fusion

• Combine detections from all feature levels.

• Apply class-specific confidence thresholding.

• Output final detection set $D* = d_i | d_i = (b_i, c_i, s_i), s_i > \tau_c$.

4. Experiment

4.1 *Benchmark and metrics*

VisDrone is a benchmark dataset specifically designed for drone vision tasks, containing a large number of images and videos captured by drones, covering various tasks such as object detection, tracking, and crowd counting. Its characteristics include diverse scenes and detailed annotations, making it particularly suitable for evaluating small object detection algorithms. Using the VisDrone dataset for the experiments in this chapter has several advantages: The data set has high-resolution features, which can capture more details, providing high-resolution images that clearly present the characteristics of small targets, facilitating algorithm recognition. It enhances robustness in complex scenes. The dataset includes diverse complex scenes with high challenges, helping to improve the adaptability of the algorithm. It has rich annotations and evaluation standards, providing detailed annotations and unified evaluation metrics to ensure the accuracy and reliability of small object detection performance. The benchmark enables comprehensive evaluation of object detection and segmentation algorithms using standard metrics, including Average Precision (AP), AP_{50}, and AP_{75}. AP quantifies the area under the Precision-Recall curve, while AP_{50} and AP_{75} denote the average precision at Intersection over Union (IoU) thresholds of 50% and 75%, respectively. Numerous state-of-the-art object detection models have been benchmarked on VisDrone, solidifying its role as a foundational dataset in the field.

4.2 *Experiment parameters*

The backbone network of the model uses a pre-trained ResNet-50, which can effectively extract image features. To improve training stability and computational efficiency, some early network layers are frozen, and batch normalization is enabled in evaluation mode. The output channels of the Feature Pyramid Network (FPN) are set to 6 to adapt to the multi-scale target characteristics in the dataset, promoting effective fusion of features at different levels. In the design of the detection head, the anchor box generation strategy is optimized, including three aspect ratios and multiple stride settings, and Focal Loss is used as the classification loss, significantly improving the detection performance of small targets and difficult samples.

In terms of training strategy, the optimizer chosen is Stochastic Gradient Descent (SGD), with an initial learning rate set to 0.01, and the learning rate is gradually reduced in the later stages of training through multi-stage adjustments to accelerate model convergence. To avoid gradient anomalies and enhance generalization ability, the gradient clipping threshold is set to 32, and the weight decay coefficient is 0.0001. For data processing, input images are uniformly resized to a resolution of 1,333 × 800, maintaining the original aspect ratio, and data diversity is enhanced through random flipping. The batch size is set to 8, and the training epochs are 100.

The model evaluation follows the COCO evaluation protocol, comprehensively measuring model performance by calculating the mean Average Precision (mAP) of bounding boxes and conducting per-category analysis. In the testing configuration, the maximum number of detection boxes is limited to 1000, and the IoU threshold for Non-Maximum Suppression (NMS) is set to 0.5 to balance detection precision and recall.

4.3 *Comparison with state-of-the-art methods*

This experiment aims to validate the performance of the proposed sparse region extraction-based object detection method on real-world datasets and compare it with the RetinaNet model. The evaluation is conducted on the VisDrone dataset, which features diverse object categories, varying scales, and high complexity, making it ideal for assessing multi-scale object detection capabilities. The experimental results are summarized in Table 1, which compares detection performance across models using standard metrics (AP, AP_{50}, AP_{75}). The data demonstrates that our model outperforms the RetinaNet baseline across all metrics, achieving improvements of + 2.9 AP, + 6.6 AP_{50} (14.7% increase), and + 1.6 AP_{75}, confirming its effectiveness. Furthermore, we observe that dense computation across high-resolution feature layers—where many regions lack small objects—can degrade model performance. By selectively processing high-probability regions, our approach enhances predictive accuracy while maintaining efficiency. Moreover, SRE-Ret achieved either

optimal or competitive performance compared to other state-of-the-art models, including QueryDet, ClusDet, DetectoRS, and CEASC.

Table 1. Evaluation results on the VisDrone validation set

	AP	AP_{50}	AP_{75}
Retinanet [15]	26.2	44.9	27.1
QueryDet [30]	28.3	48.1	28.8
ClusDet [32]	26.7	50.6	24.7
DetectoRS [33]	29.4	49.3	30.2
CEASC [34]	28.7	50.7	28.4
SRE-Ret	29.1	51.5	28.7

Table 2 further demonstrates the performance of the SRE-Ret model across 10 object categories. The proposed method achieves higher mean Average Precision (mAP) in categories such as pedestrians, people, bicycles, cars, vans, tricycles, buses, and motorcycles, with particularly notable improvements in categories dominated by small targets (e.g., pedestrians, people, and motorcycles). However, in categories with fewer small targets, such as trucks and awning-tricycles, the mAP of the proposed method exhibits a slight decline.

Table 2. Evaluation results of metrics for each category in VisDrone

Category	Retinanet	SRE-Ret
pedestrian	0.164	0.218
people	0.231	0.425
bicycle	0.090	0.090
car	0.502	0.536
van	0.280	0.296
truck	0.224	0.205
tricycle	0.132	0.146
awning-tricycle	0.090	0.082
bus	0.354	0.392
motor	0.161	0.201

Figure 8 presents partial detection results of the SRE-Ret model on the VisDrone validation dataset, with the left half displaying ground truth and the right half showing detection outputs. From these images, it is evident that the model successfully identifies the majority of small targets in small-target detection tasks. Nevertheless, in anchor-based object detection approaches, we observe the presence of redundant bounding boxes in scenes with densely packed small targets. The model generates multiple highly overlapping detection boxes, often corresponding to the same target, which introduces redundancy in the detection results and subsequently reduces both detection accuracy and efficiency. This issue may, to some extent, be attributed to hyperparameter settings during testing, potentially leading to an excessive number of fixed anchor boxes. This phenomenon suggests that the model still exhibits limitations when processing high-density

small-target scenarios, particularly in the design of the anchor box generation mechanism and the subsequent processing stages, where there remains considerable room for optimization.

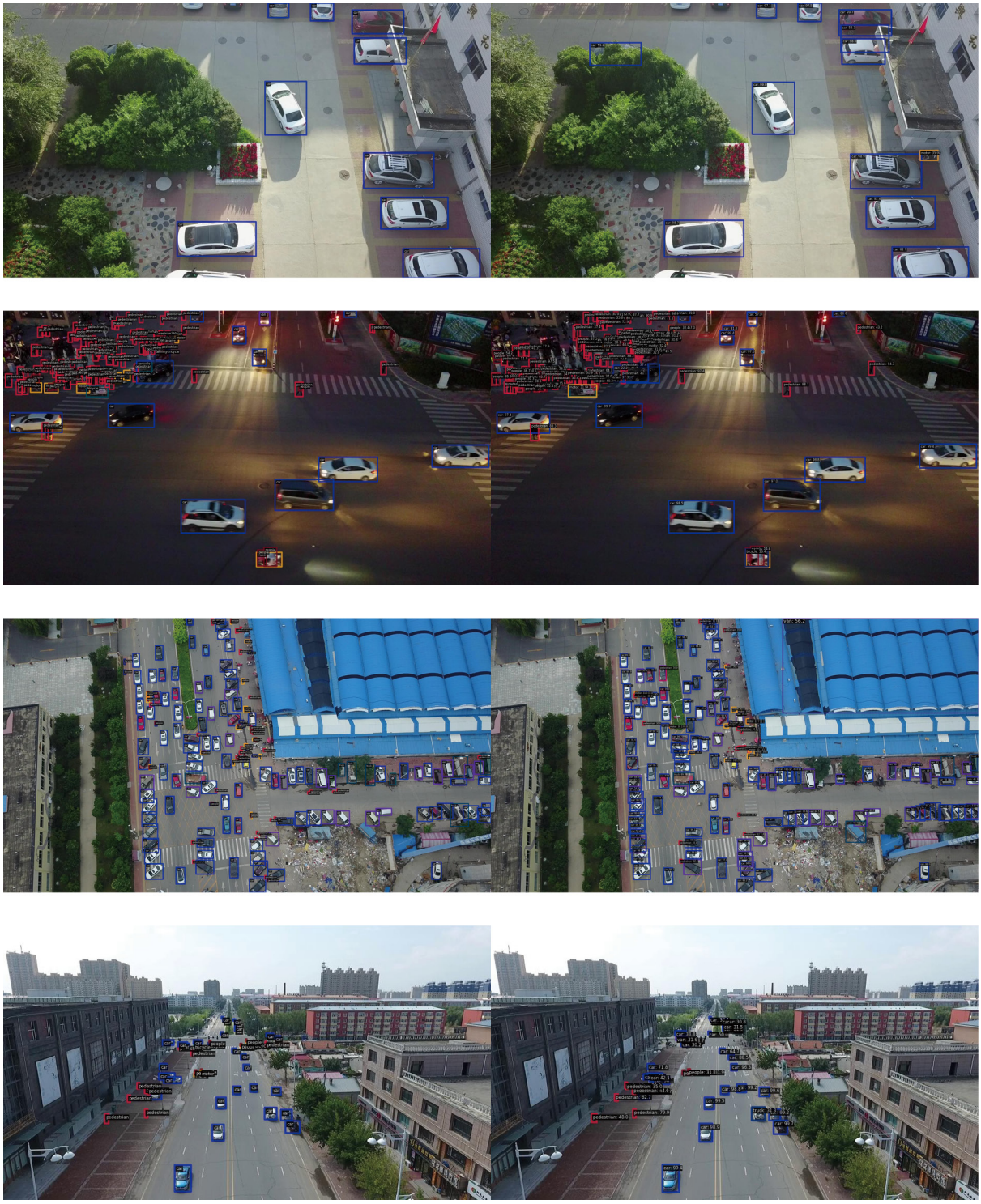

Figure 8. Partial detection results of the SRE-Ret model

Next, we conducted comparative experiments on comparison of computational efficiency (FLOP and inference time) across SRE-Net and RetinaNet.

Table 3. Comparison of computational efficiency

	FLOPs (G)	Inference Time (ms)
Retinanet [15]	97.2	45
Swin-T [21]	145.3	72
SAM2 (ViT-B) [35]	256	120
SRE-Ret	89.5	38

Table 3 shown that compared to Retinanet, SRE reduces FLOPs by about 8% by skipping computation on low-saliency regions and speeds up about 15.6% on Inference Time by avoiding processing about 20-30% of low-probability regions in high-resolution feature maps, directly reducing computation. Compared to Swin-T and SAM2 (ViT-B), SRE Net achieved better results in FLOPs and Inference Time indicators.

To improve model robustness, we analyze false negatives (missed detections) and false positives (incorrect detections).

Table 4. Results of False Cases

Error Type	Frequency	Primary Cause	mAP Impact
False Negatives	55%	Small objects (< 20px)	-7.2%
False Positives	45%	Background clutter	-5.8%

On VisDrone, small objects (< 20px) contribute to about 60% of false negatives (objects present in the image but not detected by the model) and background noise causes about 30% of false positives (detections where no object exists or the class is wrong) in urban scenes. Table 4 shown that frequencies of false negatives and false positives are 55% and 45% respectively. Correspondingly, their impact on mAP decreased by 1% and 2%, respectively.

4.4 *Ablation studies*

The Sparse Region Extraction (SRE) module identifies and processes high-probability target regions in high-resolution feature maps before forwarding them to the Transformer-based detection head for final prediction. As the primary objective of SRE is target region localization rather than category classification, we performed a comprehensive ablation study to investigate the impact of channel dimension in the SRE module.

Table 5 demonstrates the effect of varying channel dimensions on model performance. Our experiments reveal an optimal configuration at 64 channels, beyond which model performance degrades progressively. This phenomenon can be attributed to two key factors: (1) excessive channel dimensions amplify backpropagation effects on high-resolution layers, disrupting multi-scale feature fusion across the network, while (2) insufficient channels (below 64) constrain the model's learning capacity. The 64-channel configuration achieves an optimal balance, maintaining strong representational power while minimizing interference with other network layers.

Table 5. The results of different channels in the SRE module on the Visdrone dataset

	channel	*AP*	AP_{50}	AP_{75}
SRE-Ret	32	29.0	50.9	28.3
SRE-Ret	64	29.1	51.5	28.7
SRE-Ret	128	28.8	50.5	28.1
SRE-Ret	256	28.8	50.6	28.3

Table 6 presents a comprehensive comparison of the SRE-Ret model's computational performance and parameter efficiency. While the integration of the sparse region extraction module increases the total parameter count, the selective processing of only high-probability regions in the detection head significantly reduces computational overhead. This architectural optimization yields faster processing times for identical batch sizes compared to the baseline model. At the optimal 64-channel configuration, SRE-Ret demonstrates superior inference speed while maintaining accuracy. However, increasing the channel count beyond this optimum leads to greater computational demands in the SRE module, consequently diminishing the overall processing speed advantage.

Table 6. Comparison of Model Performance and Parameter Quantity

	channel	Params (M)	Overall Batch FPS (batches/s)	Average single batch processing time (ms/batch)	Batch Size
Without SRE	-	36.517	33.5	29.9	8
SRE-Ret	32	39.165	33.3	30.0	8
SRE-Ret	64	39.297	34.7	28.8	8
SRE-Ret	128	39.610	32.1	31.2	8
SRE-Ret	256	40.433	32.9	30.4	8

5. Conclusion

This paper comprehensively explores the design and implementation of an object detection method termed SRE-Ret based on sparse region extraction, aiming to address the challenges of high computational complexity and insufficient efficiency faced by traditional object detection techniques when processing high-resolution images, while also improving the recognition accuracy for small-sized targets. The method acquires enhanced features at multiple hierarchical resolutions from the image to be detected through a feature extraction module and a feature fusion module. Subsequently, sparse region extraction technology is applied to the high-resolution feature maps to select key regions and extract prior boxes for precise detection of small targets. For low-resolution features, prior boxes are directly selected from the entire feature map to achieve global target detection. Finally, the results are optimized through a non-maximum suppression module to remove redundant boxes and output the final detection results. The technical core of this invention lies in effectively reducing the computational burden through a sparse region extraction strategy while maintaining sensitivity to small targets, thereby balancing detection accuracy and efficiency. Experimental validation on the VisDrone dataset shows that, compared to the RetinaNet model, this method achieves significant improvements in overall mean average precision mAP, AP_{50}, AP_{75}, and the small target detection metric mAP_S. It particularly excels in small target detection tasks, demonstrating its potential for application in fine-grained detection scenarios. However, the experiments also reveal that the method's performance in detecting medium and large targets is relatively weaker, suggesting that future research could further optimize the fusion and processing strategies of multi-scale features to achieve more comprehensive

performance improvements. SRE-Ret can be used for feature extraction in computer vision, which can be used to extract key feature points in images, and these points can serve as the basis for subsequent processing such as face recognition and scene understanding. In medical image processing, SRE-Ret can be used to detect and analyze lesion edges, vascular boundaries, etc., to assist doctors in disease diagnosis and treatment planning. When processing satellite images or aerial photographs, SRE-Ret can be used to detect terrain features, roads, buildings, etc., which is crucial for map making, resource management, and environmental monitoring. The object detection scheme of SRE-Ret provides an innovative and practical technical approach for the field of object detection, laying a solid foundation for subsequent research.

Acknowledgement

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2023C01143, the National Natural Science Foundation of China under Grant 62072121 and the Zhejiang teaching reform project of undergraduate colleges and universities under Grant JG20220744.

Data availability

The data used in this study is available for request.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Girshick R, Donahue J, Darrell T, Malik J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015; 38(1): 142-158. Available from: http://doi.org/10.1109/TPAMI.2015.2437384.

[2] Girshick R. Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*. Santiago, Chile; 2015. p.1440-1448.

[3] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017; 39(6): 1137-1149. Available from: http://doi.org/10.1109/TPAMI.2016.2577031.

[4] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*. Venice, Italy; 2017. p.2961-2969.

[5] Cai Z, Vasconcelos N. Cascade R-CNN: delving into high quality object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA; 2018. p.6154-6162.

[6] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA; 2017. p.2117-2125.

[7] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA; 2016. p.779-788.

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA; 2017. p.7263-7271.

[9] Redmon J, Farhadi A. YOLOv3: an incremental improvement. *arXiv:180402767*. 2018. Available from: http://doi.org/10.48550/arXiv.1804.02767.

[10] Bochkovskiy A, Wang CY, Liao HY. YOLOv4: optimal speed and accuracy of object detection. *arXiv200410934*. 2020. Available from: http://doi.org/10.48550/arXiv.2004.10934.

[11] Zhang J, Zhang JS, Zhou KX, Zhang YH, Chen HD, Yan XY. An improved YOLOv5-based underwater object-detection framework. *Sensors*. 2023; 23(7): 3693. Available from: http://doi.org/10.3390/s23073693.

[12] Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications. *arXiv220902976*. 2022. Available from: https://doi.org/10.48550/arXiv.2209.02976.

[13] Wang CY, Bochkovskiy A, Liao HY. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. New Orleans, LA, USA; 2022. p.7464-7475.

[14] Kim JH, Kim N, Won CS. High-speed drone detection based on YOLO-v8. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Rhodes Island, Greece; 2023. p.1-2.

[15] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: *Proceedings of the IEEE International Conference on Computer Vision*. Venice, Italy; 2017. p.2980-2988.

[16] Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p.4171-4186.

[17] Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. *OpenAI*. 2018. Available from: https://gwern.net/doc/www/s3-us-west-2.amazonaws.com/d73fdc5ffa8627bce44dcda2fc012da638ffb158.pdf [Accessed 10th May 2025].

[18] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: *Proceedings of the European Conference on Computer Vision*. Glasgow, UK: Springer; 2020. p.213-229.

[19] Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable DETR: deformable transformers for end-to-end object detection. *arXiv201004159*. 2020. Available from: http://doi.org/10.48550/arXiv.2010.04159.

[20] Yao Z, Ai J, Li B, Zhang C. Efficient DETR: improving end-to-end object detector with dense prior. *arXiv:210401318*. 2021. Available from: http://doi.org/10.48550/arXiv.2104.01318.

[21] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal, Canada: IEEE; 2021. p.10012-10022.

[22] Wang W, Xie E, Li X, Fan DP, Song K, Liang D, et al. Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal, Canada: IEEE; 2021. p.568-578.

[23] 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. *arXiv220303605*. 2022. Available from: http://doi.org/10.48550/arXiv.2203.03605.

[24] Roh B, Shin J, Shin W, Kim S. Sparse DETR: efficient end-to-end object detection with learnable sparsity. In: *Proceedings of the International Conference on Learning Representations*. Virtual Event: ICLR; 2022. .

[25] Srinivas A, Lin TY, Parmar N, Shlens J, Abbeel P, Vaswani A. Bottleneck transformers for visual recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual Event: IEEE; 2021. p.16519-16529.

[26] Peng Z, Huang W, Gu S, Xie L, Wang Y, Jiao J, et al. Conformer: local features coupling global representations for visual recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal, Canada: IEEE; 2021. p.367-376.

[27] Dai Z, Liu H, Le Q, Tan M. Coatnet: marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*. 2021; 34: 3965-3977. Available from: http://doi.org/10.48550/arXiv.2106.04803.

[28] Mehta S, Rastegari M. MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv211002178*. 2021. Available from: http://doi.org/10.48550/arXiv.2110.02178.

[29] Singh B, Najibi M, Davis L. Sniper: efficient multi-scale training. In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. Montréal, Canada; 2018. p.31.

[30] Yang C, Huang Z, Wang N. QueryDet: cascaded sparse query for accelerating high-resolution small object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. New Orleans, LA, USA: IEEE; 2022. p.13668-13677.

[31] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: transformers for image recognition at scale. *arXiv201011929*. 2020. Available from: http://doi.org/10.48550/arXiv.2010.11929.

[32] Yang F, Fan H, Chu P, Blasch E, Ling H. Clustered object detection in aerial images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Seoul, Korea (South): IEEE; 2019. p.8311-8320.

[33] Qiao S, Chen LC, Yuille A. Detectors: detecting objects with recursive feature pyramid and switchable atrous convolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual Event: IEEE; 2021. p.10213-10224.

[34] Du B, Huang Y, Chen J, Huang D. Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vancouver, Canada: IEEE; 2023. p.13435-13444.

[35] Ravi N, Gabeur V, Hu YT, Hu R, Ryali C, Ma T, et al. SAM 2: segment anything in images and videos. *arXiv240800714*. 2024. Available from: https://doi.org/10.48550/arXiv.2408.00714.