Feature Pyramid Optimization-Based Small Object Detection in UAV Imagery

by

Zening Wang

A thesis submitted to University of Ottawa
in partial fulfillment of the requirements for the
Master of Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Unmanned Aerial Vehicles (UAVs) are increasingly used in key applications such as surveillance, search and rescue. However, due to the small scale of objects, cluttered backgrounds, and inherent information loss, accurate object detection from a high-altitude perspective is still a major challenge. Although two-stage detectors can provide high accuracy, their high computational costs are not suitable for real-time edge deployment. In contrast, the most advanced single-stage detectors, such as YOLOv10, often fail to capture small objects because the deep pyramid structure loses key spatial details in the downsampling process. In order to overcome these limitations, this paper proposes a systematic optimization framework for small object detection in high-altitude imagery. We first introduce FemtoDet-P2, which combines the MambaOut backbone network with a high-resolution P2 detection head. As a high-precision baseline, this architecture verifies that strengthening early feature extraction is crucial to solving the problem of feature vanishing, even if structural redundancy is introduced. Based on this, we propose LSCNet to balance the trade-off between detection accuracy and computational efficiency. LSCNet eliminates the redundancy in the baseline model and enhances multi-level feature fusion by constructing a shallow feature cascade and introducing a lightweight attention mechanism. This lightweight design ensures that advanced deep learning models can be efficiently deployed on resource-constrained UAV platforms without compromising their ability to identify small objects in complex environments.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Amiya Nayak, for providing me with guidance, advice, and support throughout my master's studies. Finally, I would like to thank my family, who have continuously supported me both emotionally and unconditionally.

Contents

List of Tables

List of Figures

Acronyms

AP Average Precision.

CNN Convolutional Neural Network.

COCO Common Objects in Context.

CSP Cross Stage Partial.

CSPNet Cross Stage Partial Network.

DETR DEtection TRansformer.

DyHead Dynamic Head.

ECA Efficient Channel Attention.

ECAOK ECA-enhanced Omni-Kernel.

ERF Effective Receptive Field.

FN False Negatives.

FP False Positives.

FPN Feature Pyramid Networks.

FPS Frames Per Second.

GELAN Generalized Efficient Layer Aggregation Network.

GELU Gaussian Error Linear Unit.

GFLOPs Giga Floating-point Operations.

GT Ground Truth.

HOG Histograms of Oriented Gradients.

ILSVRC ImageNet Large Scale Visual Recognition Challenge.

IoU Intersection over Union.

LoG Laplacian-of-Gaussian.

LSCNet Lightweight Shallow Feature Cascade Network.

mAP Mean Average Precision.

MLP Multi-Layered Perceptron.

NMS Non-Maximum Suppression.

PANet Path Aggregation Network.

PGI Programmable Gradient Information.

R-CNN Region-based Convolutional Neural Networks.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

RPN Region Proposal Network.

RT-DETR Real-Time DEtection TRansformer.

SAOK Small-target Aware Omni-Kernel.

SPPF Spatial Pyramid Pooling - Fast.

SSD Single Shot MultiBox Detector.

SSM State Space Models.

SVM Support Vector Machines.

TN True Negatives.

TP True Positives.

UAV Unmanned Aerial Vehicle.

UAVDT UAV Detection and Tracking.

Vim Vision Mamba.

ViT Vision Transformer.

YOLO You Only Look Once.

Chapter 1

Introduction

1.1 Motivation & Challenge

Unmanned aerial vehicles are becoming more and more common in a number of applications, such as precision agriculture, traffic monitoring, search, rescue, and surveillance. In these tasks, the system needs to accurately distinguish motorcycles, pedestrians and cars from an aerial perspective to optimize traffic flow. Similarly, in maritime search and rescue operations, it is vital to identify life jackets or swimmers on the vast sea surface to save lives. These objects usually appear only as small points in the image, so detecting them is a significant task.

However, due to a number of intrinsic challenges, object recognition from aerial perspectives continues to be a major hurdle in computer vision. Objects captured by UAV platforms often exhibit extremely small scales. These objects also experience significant information loss due to motion blur, ambient interference, and changing lighting conditions. Accurate detection is especially difficult because of the high-altitude viewpoint, which introduce complicated background clutter and notable size fluctuations.

Historically, object detection has relied heavily on manually designed features, such as Haar cascades[7] or Histograms of Oriented Gradients (HOGs)[8], combined

with traditional classifiers such as Support Vector Machines (SVM). Although these methods lay the foundation for computer vision, they rely on manual feature engineering. The emergence of convolutional neural networks (CNNs) has completely changed this situation. They can automatically learn hierarchical features directly from the original pixel data, without manual design, and significantly improves the detection accuracy in complex environments[9].

Although two-stage detectors like Faster R-CNN [10] have high accuracy, their huge computational cost and slow inference speed are not suitable for real-time processing on drones. Similarly, although Transformer-based models such as RT-DETR-R18 [11] have global receptive fields, they usually perform poorly in small object detection because their architecture may lose important spatial details during the downsampling process. In contrast, single-stage detectors, especially the YOLO (You Only Look Once) series, are more suitable for edge deployment by balancing speed and accuracy [12]. Among them, YOLOv10 [13] reduces inference latency by eliminating the non-maximum suppression (NMS) post-processing required by traditional YOLO, rendering it an ideal choice for resource-constrained environments. However, although the computational efficiency of YOLOv10 is very high, the results of recognizing small objects in aerial images taken by UAVs are not satisfactory. This is because deep pyramid models lack the ability to extract and utilize early information suitable for small objects[14]. YOLOv10 is mainly designed for natural images (such as the COCO dataset [15]), where objects are clear and located in the center of the image. Our empirical analysis shows that the model wastes a large amount of computational resources on the deep pyramid to extract high-level semantic abstractions, which are largely redundant for tiny, pixel-limited objects typical in drone perspectives.

1.2 Research Objectives

The main objective of this thesis is to systematically study the optimization of small object detection for high-altitude UAV platforms, propose a lightweight solution based on commonly used YOLOv10 architecture, and compare it against the state-of-the-art methods[2, 16, 17].

1.3 Thesis Contributions

The two main contributions of this work are as follows:

- We utilize MambaOut as the backbone to enhance the model design for early feature recognition. This model forms the basis for the optimization strategies that will be used in developing a lightweight solution for small object detection.

- We introduce LSCNet, a specialized architecture designed for UAV-based small object detection. Based on our empirical analysis, we identified that deep feature extraction layers are redundant for small objects. This streamlined framework is supported by three core innovations: the LoGStem module, the SAOK module, and the DyHead module. These components work together to enable LSCNet to achieve a balance between inference efficiency and high-precision detection. We perform a comprehensive evaluation and comparison on VisDrone2019 [18] and UAVDT [19] benchmark datasets, benchmarking our model against state-of-the-art lightweight models. The thesis work has resulted in the following journal paper:

  Z. Wang, A. Nayak, "LSCNet: A Lightweight Shallow Feature Cascade Network for Small Object Detection in UAV Imagery", Future Internet 17 (12), 568, 2025.

1.4 Thesis Organization

The remainder of this thesis is organized as follows.

- Chapter 2 provides an overview of fundamental concepts relevant to this research, including Neural Networks, the evolution of YOLO, challenges in aerial small object detection, LoG edge detection, and Transformers.
- Chapter 3 reviews the related work and literature. It discusses the evolution of feature pyramid networks, analyzes the specific challenges of small object detection in aerial imagery, and surveys recent advancements in attention mechanisms and State Space Model (SSM).
- Chapter 4 presents FemtoDet-P2, an architecture that investigates the "efficient backbone + heavy neck" design strategy.
- Chapter 5 presents LSCNet, the core contribution of this thesis. Based on the research results in Chapter 4, this chapter introduces the Shallow Feature Cascade strategy. This strategy removes the redundant deep pyramid layer and integrates the LoGStem, SAOK and DyHead modules, thus achieving the optimal balance between detection accuracy and model efficiency.
- Chapter 6 presents the comprehensive evaluation of the proposed models. It details the implementation environment and evaluation metrics, and discusses the experimental results on the VisDrone2019 and UAVDT benchmarks, including ablation studies and comparisons with state-of-the-art lightweight models.
- Chapter 7 concludes this thesis with a summary of key findings and discusses possible future research directions.

Chapter 2

Background

2.1 Neural Network

Neural Network is a machine learning technique that uses artificial neurons to simulate how biological neurons in the human brain work. The basic element of each neural network is the artificial neuron which consists of three components: weights and biases, summation function, and activation function.

As shown in Figure 2.1, each neuron is embedded with one summation function and one activation function; the summation function is usually a linear regression function with weight and bias, while the activation function provides non-linearity transformation for the summation output. Suppose a given input sample has n features, denoted by a vector $X = [x_1, x_2, \ldots, x_n]^T$. Each input feature x_j is associated with a weight w_j, and the neuron has an inherent bias term b. The linear summation function for this neuron is defined as:

$$y = \sum_{j=1}^{n} w_j x_j + b \tag{2.1}$$

A neural network is made up of layers of interconnected nodes (neurons). The

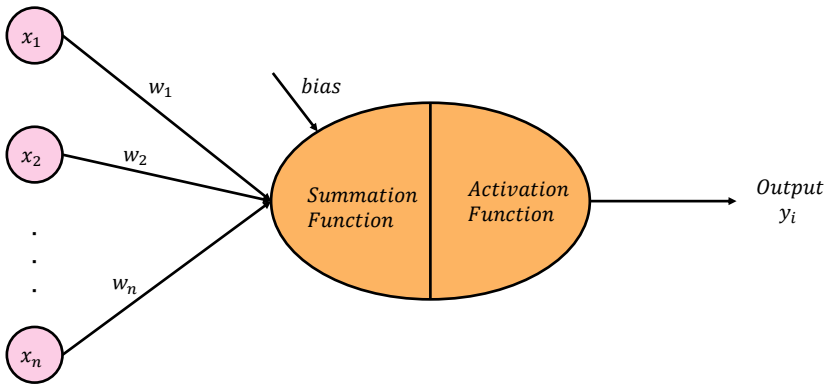


Figure 2.1: Structure of an Artificial Neuron

signal obtained by the linear summation is fed forward into a non-linear activation function.

Figure 2.2 illustrates the architecture of a simple neural network. In this Multi-Layer Perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer is responsible for collecting input patterns. The output layer contains classifications or output signals that may be mapped to by input patterns. In the middle, there is a hidden layer. Hidden layers adjust the input weightings until the neural network's margin of error is as little as possible by experiments.

2.2 Supervised Neural Network Models

Using a data set to train the supervised learning algorithm, the Multi-Layer Perceptron (MLP) can learn a function $f(X) : R^m \rightarrow R^o$, where m represents the number of dimensions for input and o shows the number of dimensions for output.

The MLP can learn a non-linear function for classification or regression tasks. Compared to logistic regression, it can discover the hidden relationships between

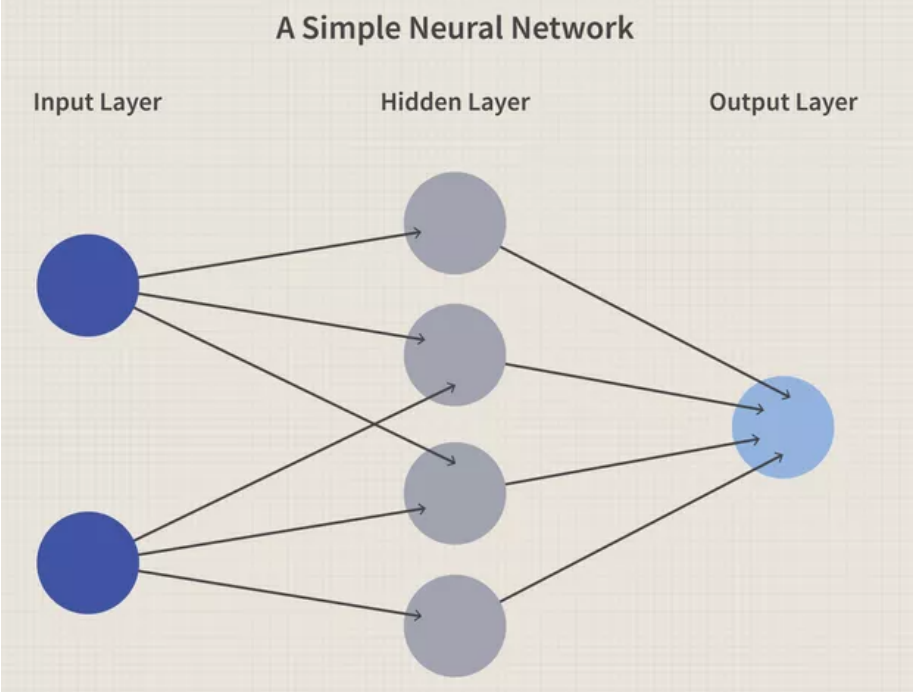


Figure 2.2: A Simple Neural Network with Layers[3]

input features and the output through its hidden layers. As shown in Figure 2.2, the input layer contains the features, followed by non-linear layers in the middle.

The benefits of MLP are significant; it is capable of modeling complex non-linear relationships. However, there are some drawbacks: MLP is sensitive to feature scaling and requires tuning a large number of hyperparameters, such as the number of hidden neurons, layers, and iterations.

2.3 Convolutional Neural Networks

The emergence of Convolutional Neural Networks (CNNs) revolutionized object detection. Inspired by the human visual cortex, CNNs automatically learn hierarchical features directly from raw pixel data, eliminating the need for manual feature engineering.

A typical CNN architecture shows in Figure 2.3, which consists of two primary stages: feature extraction and classification. In the feature extraction phase, the

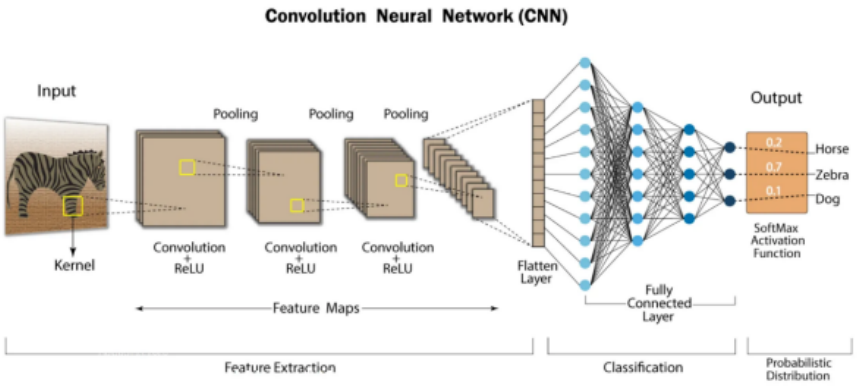


Figure 2.3: The structure of CNN[4]

convolution layers apply kernels to capture local patterns such as edges and textures, followed by ReLU activation to introduce non-linearity, and pooling layers to reduce spatial dimensions while preserving essential information. The final output is obtained through a SoftMax activation function, thus providing a probabilistic distribution of the classes.

2.4 Cross Stage Partial Network [1]

As the depth of convolutional neural network increases, the computational burden becomes the main bottleneck that restricts its deployment on edge devices (such as UAVs). In order to solve this problem and improve inference speed, Wang et al. proposed Cross Stage Partial Network (CSPNet). CSPNet has a profound influence on the development of modern object detectors, especially YOLO series. The core design concept of CSPNet is to solve the problem of duplicate gradient information in the deep network. In traditional convolutional networks (such as ResNet or DenseNet), the backpropagation of gradients often involves a lot of repeated calculations, resulting in a waste of resources. CSPNet optimizes this process through a cross-stage partial connection strategy. Specifically, it divides the feature map of the basic layer into two parts: one part of the feature map continues to extract deep features through the original path of the convolution layer, and the other part of

the feature map directly skips the middle layer and concatenates with the processed features at the end of the phase. This design brings three key advantages: Gradient flow is enhanced: by separating and merging feature paths, the gradient information of different paths is more abundant, and the feature learning capability is improved. Significantly reduces the amount of computation: because only part of the feature map enters the computationally intensive convolutional block, the number of parameters and FLOPs of the model are significantly reduced, which is very suitable for resource constrained UAV platforms. The accuracy is maintained: the experiment proves that CSPNet does not sacrifice the detection accuracy while maintaining a lightweight architecture. Because of this, the design concept of CSPNet has become the cornerstone of backbone network design in YOLOv4, YOLOv5 and subsequent versions (including the architecture referenced by the baseline model of this study), providing the necessary architecture foundation for real-time small object detection.

2.5 Object Detection

Object detection is a fundamental task in computer vision that involves identifying and localizing objects of interest in an image or video. Unlike image classification, which assigns a single or multiple labels to the entire image, object detection provides class labels and spatial coordinates (usually bounding boxes) for multiple objects. This technology supports a wide range of applications, including autonomous driving, surveillance, medical imaging, and augmented reality.

There are four major categories of image recognition tasks in computer vision:

- Classification: Given a picture or a video, determine what category of objects it contains.
- Localization: Locate the location of this object.
- Detection: Locate the location of the object and know what the object is.

- Segmentation: It is divided into instance segmentation (Instance-level) and scene segmentation (Scene-level).

Our task is the third type of object detection, detecting the location and category of the object in the picture. In the history of object detection, the early methods relied on hand-crafted features, such as Haar cascades or histograms of oriented gradients (HOG), combined with classifiers such as support vector machines (SVMs). In 2014, R-CNN[20] was the first to introduce a region-based method with CNN features, significantly improving the accuracy of object detection. Generally, these tasks are divided into two-stage and one-stage methods.

Two-stage object detection technology, such as R-CNN[21] and its variants (Fast R-CNN, Faster R-CNN[22]), Faster R-CNN first uses a Region Proposal Network (RPN) in the first stage to generate potential object candidate regions. In the second stage, these candidate areas are then sent to a deep convolutional network for feature extraction, and the category and precise location of the object are finally determined. The advantage of dual-stage object detection is that through two-stage processing, objects in the image can be more accurately located and classified. However, the disadvantages are also obvious. When generating a large number of regional proposals in the first stage, computing resources are consumed greatly, which may not be efficient enough in real-time applications.

Single-stage object detection has gained widespread application in recent years due to its real-time processing capabilities. For example, YOLO (You Only Look Once)[23] and SSD (Single Shot MultiBox Detector)[24] predict the category and location of the object directly from the image through a separate neural network model, omitting the region proposal stage in the traditional two-stage method. The main advantage is their efficiency, they enable faster inference without sacrificing too much accuracy, which makes them ideal for applications that require real-time processing.

In recent years, significant progress has been made in object detection technology, including Transformer based models (such as DETR [25]), which can improve the semantic processing ability of models according to the long-range dependency modeling capability of self-attention, so as to improve the detection accuracy of models. With the continuous development of object detection technology, small object detection and handling occlusion in complex environments are still worthy of further exploration.

2.6 Small Object Detection

Small object detection is one of the most challenging subtasks in computer vision, especially in the UAV aerial photography scene. According to the general definition of MS COCO dataset, objects smaller than 32×32 pixels are classified as Small Objects. From a UAV perspective, due to the high flying altitude, the size of ground vehicles and pedestrians is often much smaller than this standard, and may even be less than 16×16 pixels.

The main challenges are as follows: Feature Vanishing, Low Signal-to-Noise Ratio and Position Sensitivity. Modern convolutional neural networks usually include multiple downsampling operations, such as P5 layer with Stride=32. For a small object of 16×16 pixels, after five downsampling, there may be less than one pixel of information left in the deep feature map, or even disappear completely. Small objects are very sensitive to the regression error of bounding box. Small pixel-level deviation will lead to significant performance degradation in IoU calculation. Also, small objects cover very few pixels, and their appearance features are easily submerged by complex background textures, such as vegetation, road markings, which makes it difficult for the detector to distinguish between foreground and background.

Therefore, extracting features from the edges of small objects is crucial.

2.7 Fundamentals of Edge Detection

Edge detection is a basic operation in image processing, which aims to identify areas with sharp changes in image brightness, which usually correspond to the contours of objects. In UAV aerial images, because the object size is very small and vulnerable to motion blur, clear edge features are essential for accurate object localization. The Laplacian is a second-order differential operator, which is used to detect intensity step changes in images. For the image $I(x, y)$, its Laplacian definition is

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{2.2}$$

Laplacian operator is very sensitive to noise, and its direct application to aerial images with high noise may result in a large number of false edges. Laplacian of Gaussian (LoG): In order to solve the problem of noise sensitivity, Marr and Hildreth proposed the Gaussian Laplacian operator [26]. The method first uses a Gaussian function to smooth and denoise the image, and then uses the Laplacian operator to extract the edge. This combines the noise reduction ability of Gaussian smoothing and the edge localization ability of second-order differential. The mathematical form of the LoG operator can be expressed as:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.3}$$

When processing low quality or blurred UAV images, the LoG operator can effectively enhance the edge features of the object and suppress high-frequency background noise, which is also the theoretical basis for introducing the LoGStem module in the shallow feature extraction phase of this study.

2.8 Evolution of the YOLO Series

The YOLO series exemplifies the advancement of single-stage detectors:

- **YOLOv1**: Introduced the regression-based approach, offering speed but lower accuracy.
- **YOLOv2**: Incorporated anchor boxes and multi-scale training to improve precision.
- **YOLOv3**: Utilized Darknet-53 and Feature Pyramid Networks (FPN) for enhanced multi-scale detection.
- **YOLOv4**: Integrated optimizations such as CSPNet [1] and PANet [27].
- **YOLOv5**: Improved usability and performance (unofficial release).
- **YOLOv6 and YOLOv7**: Further refined architecture and training strategies.
- **YOLOv8**: Combines CNN efficiency with Transformer-inspired attention mechanisms, achieving state-of-the-art performance.
- **YOLOv9**: Introduced Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) to address information bottleneck issues in deep networks.
- **YOLOv10**: Proposed an NMS-free training strategy via consistent dual assignments and a holistic efficiency-accuracy-driven model design, significantly reducing inference latency.
- **YOLOv11**: Refined the backbone with C3k2 blocks and integrated C2PSA attention mechanisms to achieve superior feature extraction efficiency and parameter optimization.

- **YOLOv12**: Adopts an attention-centric architecture (e.g., Area Attention), maximizing the potential of attention mechanisms within real-time constraints to surpass CNN-based limitations.

This study focuses on YOLOv10 as the baseline model for small object detection, even though the YOLO series has progressed to YOLOv11 and YOLOv12.

YOLOv8 and YOLOv10 currently serve as the most stable industrial baseline with established extensibility. We specifically investigate YOLOv10 due to its NMS-free training strategy. In small object detection scenarios, objects often appear in dense clusters. Traditional NMS tends to suppress valid adjacent objects; YOLOv10's dual assignment strategy fundamentally mitigates this issue, making it highly relevant for our research objectives.

While YOLOv11 and v12 show improvements on COCO benchmarks, their advancements focus on general feature extraction efficiency rather than addressing the specific feature vanishing challenges of small objects. Furthermore, the newest iterations lack the extensive community validation of YOLOv8 and v10.

2.9 Transformer

The Transformer architecture, initially proposed for natural language processing, relies on a self-attention mechanism to capture long-range dependencies within sequential data.[5] Unlike recurrent neural networks, Transformers process input sequences in parallel, enabling efficient training and scalability. The core component, self-attention, computes weighted relationships between all elements in a sequence, allowing the model to focus on relevant parts of the input.

As shown in Figure 2.4, each encoder layer consists of two main sub-layers: a multi-head self-attention mechanism and a feed-forward neural network, both followed by an "Add & Norm" step that applies residual connections and layer normalization. The decoder mirrors this structure but includes an additional masked multi-head attention

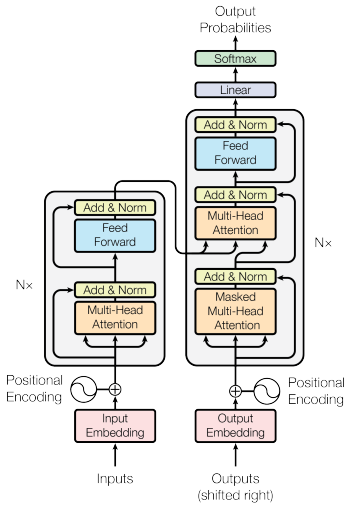


Figure 2.4: The structure of attention mechanism[5]

sub-layer to prevent attending to future tokens, ensuring auto regressive generation. The attention mechanism, central to the Transformer, computes a weighted sum of input embeddings based on their relevance, determined by query, key, and value vectors derived from the input. Specifically, for an input sequence of length n, the self-attention mechanism calculates attention scores as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where Q, K, and V are the query, key, and value matrices, and d_k is the dimension of the keys.

Positional encodings are added to the input embeddings to retain sequential information, enabling the Transformer to model long-range dependencies efficiently. The output of the decoder is passed through a linear layer and softmax to produce probability distributions over the token vocabulary.

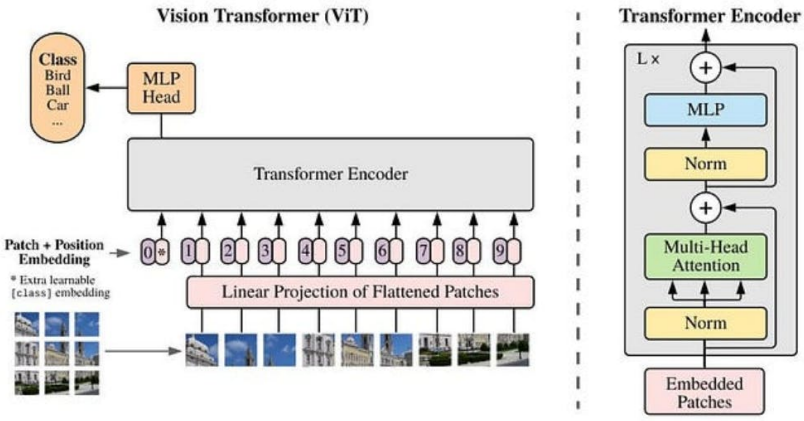


Figure 2.5: The structure of ViT[6]

2.10 Transformer in Computer Vision

The Vision Transformer (ViT)[6] is a groundbreaking technology in the field of computer vision. It adopts the Transformer architecture, which was originally used primarily for natural language processing. The core innovation is to decompose images into small 16×16 image patches, treating them as elements in a sequence, similar to words in text processing.

As illustrated in Figure 2.5, an input image is first divided into fixed-size patches, which are then flattened and linearly projected into a sequence of embeddings. A learnable class token is prepended to the sequence to capture global image information, and positional embeddings are added to retain spatial information. This sequence is fed into a stack of Transformer encoder layers, each consisting of multi-head self-attention and a multi-layer perceptron (MLP), with normalization and residual connections applied at each step.

The self-attention mechanism allows ViT to model relationships across all patches, enabling it to capture global context effectively. After processing through the L encoder layers, the class token's output is passed through an MLP head to produce the final classification probabilities. In addition to traditional attention, recent studies have also used the gating mechanism to simulate selective attention.

Chapter 3

Related Works

3.1 Challenges in Aerial Imagery and Feature Pyramids

Object detection in UAV aerial imagery requires balancing high precision for tiny targets and low computational cost for edge deployment. This is primarily due to the extreme scale variations, specific viewing angles, and complex backgrounds inherent in aerial image recognition scenarios, as well as the constraints of UAV endurance and onboard equipment. In this section, we review related literature across three dimensions: multi-scale feature representations, small object detection in aerial imagery and attention mechanisms.

The evolution of the YOLO series has progressed alongside the development of feature pyramids, from YOLOv3 [28], which first used the FPN concept, to later YOLO versions [13, 29–31] that integrated PANet's bottom-up pathway architecture. From this evolutionary trajectory, we can see how YOLO has continuously adapted to pyramidal architectures. Today, the YOLO series has become a representative example of feature pyramid networks. Most notably, YOLOv10 [13] eliminates non-maximum suppression (NMS) and introduces efficient architectural designs to minimize latency.

However, when these models developed for general tasks are applied to drone aerial scenarios, their deep structures often lose fine features due to downsampling.

To overcome the resolution limitations of generic detectors, researchers have developed specialized strategies for high-altitude aerial imagery. Early approaches such as ION [32] utilized context-aware mechanisms, called spatial recurrent neural networks, to address the issue of insufficient feature resolution. Relation Networks [33], meanwhile, enhanced feature discriminability by modeling appearance and geometric relationships. Another direction incorporates super-resolution techniques that generate fine details of small objects at higher resolutions, demonstrated by Perceptual GAN [34] and MTGAN [35].

3.2 Attention Mechanisms and Transformers

Beyond feature pyramids, recent advancements employ Transformer architectures and attention mechanisms to capture global context. Vision Transformer (ViT)[6] is a groundbreaking technology in the field of computer vision. It adopts the Transformer architecture and was originally mainly used for natural language processing. The core innovation is to decompose images into small 16×16 image patches, treating them as elements in a sequence, similar to words in text processing. These image patches are then converted to embedding vectors through linear projection and fed into a standard Transformer encoder. Furthermore, ViT introduces position embedding to preserve the position information of image patches. ViT is also widely used in object detection tasks.

In the specific domain of aerial imagery, these Transformer-based methods have shown promise but also limitations. More recently, studies have increasingly turned to Transformer-based architectures. The reason is purely CNN-based methods often suffer from limitations in processing global contextual information. In this domain, DETR [36] and UAV-DETR [37] have demonstrated robust performance by effec-

tively capturing global dependencies. To further address the specific challenges of UAV imaging, several hybrid approaches have been proposed. For instance, RingMo-Lite [38] employs a dual-branch structure combining CNNs and Transformers, and Hyneter [39] utilizes a hybrid backbone with dual-switching modules to fuse local data with global dependencies. Similarly, AST [40] introduces a pyramid structure to jointly learn local details and global dependencies at various scales. Collectively, these studies demonstrate the potential of integrating attention mechanisms with feature pyramid structures. However, these methods bring substantial computational overhead. Context modeling and GAN-based generation significantly increase inference latency, while Transformers typically demand extensive GPU memory and computing power.

To mitigate the computational cost of Transformers, lightweight attention mechanisms are preferred. Lightweight attention mechanisms resolve the conflict between feature enrichment and computational constraints. Channel attention methods, such as SENet [41] and Efficient Channel Attention (ECA) [42], adaptively recalibrate feature channels to emphasize informative representations. ECA, in particular, is highly efficient, avoiding dimensionality reduction through local cross-channel interaction. Beyond channel-wise refinement, unified frameworks like Dynamic Head (DyHead) [43] integrate scale-aware, spatial-aware, and task-aware mechanisms to coherently optimize representation across multiple dimensions.

3.3 Vision Mamba and Mambaout

The computational complexity of the traditional Transformer self-attention mechanism is quadratic with the length of the image sequence. This means that the larger the image, the greater the amount of computation and memory consumption. Vision Mamba (Vim) [44] is a vision backbone network based on bidirectional Mamba modules. Its core advantage is to reduce the complexity from quadratic to linear.

Mamba initially [45] achieved success in the field of natural language processing and was rapidly applied to computer vision. Mamba architecture demonstrates that recurrent neural network (RNN) can effectively model long sequences by compressing historical data into fixed-size states. Native Mamba is designed for language models, which is unidirectional (only the previous information can be seen). Vim introduces bidirectional design to model image sequences from both forward and reverse directions to capture bidirectional global visual context. Vim embeds the position into the sequence to enhance the spatial awareness of the model.

However, MambaOut [46] questioned the value of Vim in visual recognition tasks. There are two main arguments. First, Mamba's recurrent mechanism perfectly matches the causal relationship of language. Because in a language, the current lexical element is strictly dependent on the previous lexical element. In contrast, visual recognition is a non-causal "understanding" task. We humans can see the whole image at the same time, and the same is true of the model, and each pixel is spatially associated with its adjacent pixels in all directions, not just those preceding it in the flattened sequence. In other words, each pixel exhibits a multidirectional spatial correlation with its neighbors, rather than a strict left-to-right causal dependency. Forcing non-causal image data into the causal sequential processing flow will impose unnecessary restrictions, and the static characteristics of images cannot be utilized. Second, Mamba's state space model (SSM) memory mechanism is "lossy memory". When dealing with extremely long sequences, it exchanges linear computational complexity for a fixed size of hidden state. The authors of MambaOut calculated that the standard ImageNet input (224×224) corresponds to a very small number of tokens (about 196). At this length, the computing overhead of the Transformer is completely acceptable, without sacrificing memory integrity for efficiency. Only in high resolution tasks (more than 4000 tokens) can the sequence length be enough to make Mamba's efficiency advantage appear.

Vision Mamba essentially adds a state space model (SSM) module based on the

Gated CNN framework, while MambaOut removes the SSM component and retains the pure Gated CNN framework. Although the authors of MambaOut found that SSM is beneficial to long sequence detection tasks, in the UAV small object detection scenario, we pay more attention to the improvement of detection speed brought by Gated CNN and the high-resolution spatial information provided by P2 layer, so we choose to trade a degree of the global context modeling capability in exchange for higher efficiency and better shallow feature representation. For the small object detection task, only using Gated CNN module is enough to meet the demand, and can significantly improve the running speed. Beyond feature extraction, the Gated CNN dynamically modulates feature weights according to context, effectively suppressing background noise while enhancing relevant signals.

Chapter 4

FemtoDet-P2: YOLOv10 based Small Object Detection Framework

4.1 Introduction

Now in the field of high-altitude object detection, in order to enable edge devices such as UAVs to carry airborne models to achieve real-time object detection. Standard detectors, including the YOLO series, aggressively downsample input images to reduce computational load. While effective for large objects, this approach proves detrimental for aerial objects that occupy fewer than 16×16 pixels, as these objects physically vanish after multiple pooling operations. This is also a major challenge in current detector design. Early features provide substantial benefits for accuracy, but excessively high resolution increases computational overhead. How to balance accuracy and model computational cost is a critical problem that needs to be addressed for small object detectors, particularly for UAV-mounted models. This chapter will propose a FemtoDet-P2 model based on YOLOv10 to address these issues. The specific model structure is shown in the figure 4.1.

As we mentioned in the related work, although the Mamba (SSM) architecture solves long-range dependency, it introduces a complex hardware scanning mecha-

nism, and inference speed is not always superior on end devices. MambaOut further optimizes the original Mamba structure and provides a lightweight solution for low-resolution images. Therefore, we integrated MambaOut as the backbone network to enhance the feature extraction capability. This approach leverages the Gated CNN mechanism to improve the semantic quality of high-resolution features. Additionally, in order to retain more early features, we extend the feature pyramid to the P2 level. The newly added P2 detection head can better extract and process early features. Additional P2 layers for small object detection optimization were already being utilized in the early stages of YOLO development.

4.2 MambaOut-Femto

As mentioned in the related work, since the SSM module is removed, MambaOut is more efficient than the original Vision Mamba. Based on the architectural advantages of gated CNN, we built the FemtoDet-P2 model. We use the MambaOut Femto variant as the feature extractor. As described in the previous section, the architecture stacks gated CNN blocks in four stages. Specifically, the configuration is defined by the depth $[3, 3, 9, 3]$ and the embedding dimension $[48, 96, 192, 288]$. Unlike the standard YOLO backbone network, which usually outputs features from Phase 2 (P3), MambaOut provides a hierarchical feature pyramid starting from Phase 1. Let $I \in \mathbb{R}^{H \times W \times 3}$ be the input image. The backbone network generates a set of four multi-scale features $\{C_2, C_3, C_4, C_5\}$:

- $C_2 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 48}$: The shallowest feature map, rich in texture and edge information, is crucial for small object localization.
- $C_3 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 96}$: Standard low-level features.
- $C_4 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 192}$: Intermediate semantic features.

- $C_5 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 288}$: The deepest semantic features, possessing the largest receptive field.

The figure of FemtoDet-P2 shown as Figure 4.1.

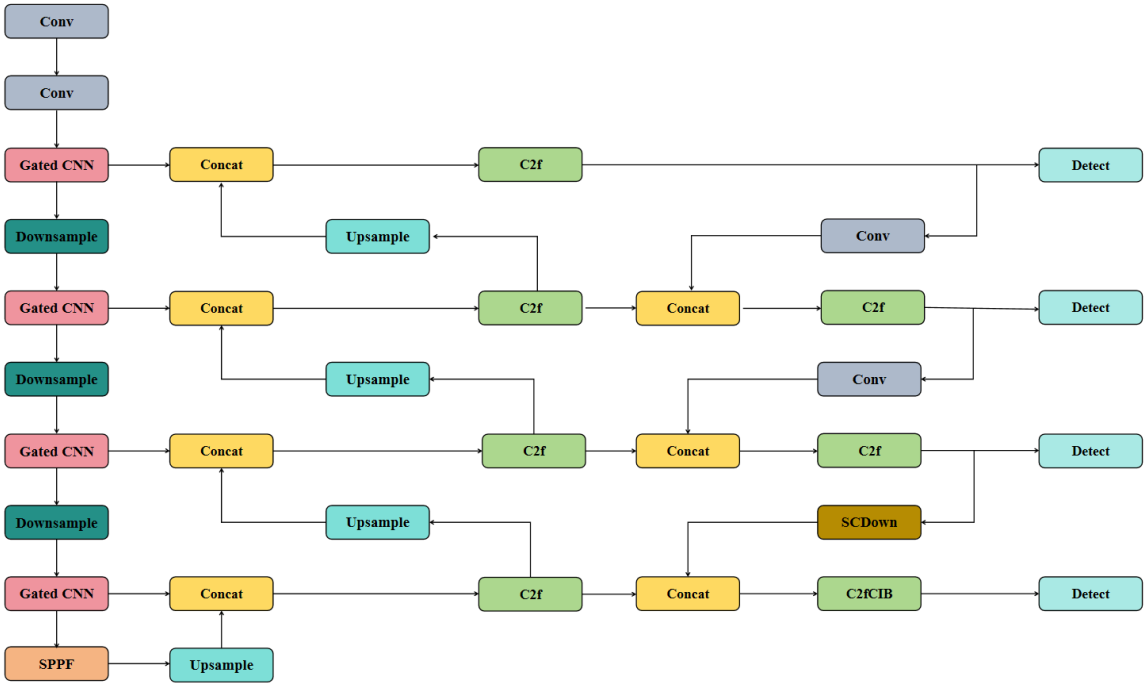


Figure 4.1: Structure of FemtoDet-P2.

4.3 Gated CNN

We introduce the Gated CNN Block into the backbone network design. Although the C2f module optimizes feature extraction through rich gradient flow, its limited 3×3 local receptive field is insufficient to capture the global context information required by small objects from a UAV perspective in shallow layers. For this reason, Gated CNN Block introduced a "gated attention" mechanism to simulate the selectivity of sequence models, and used large kernel convolution to capture wider local context while maintaining spatial focus.

For the input feature $X \in \mathbb{R}^{H \times W \times C}$, the forward propagation process of Gated CNN Block can be formalized as:

$$Y = \text{Proj}_{out}\left(\sigma(\text{Proj}_{in1}(X)) \odot \text{DepthConv}_{7\times7}(\text{Proj}_{in2}(X))\right) + X \tag{4.1}$$

This structure abandons the computationally intensive SSM module in Vision Mamba and adopts an efficient "parallel dual-stream processing mechanism" instead. According to the detection task characteristics of YOLO, this module addresses the challenges of background noise suppression and context completion for small objects through the following two key branches, show in the Figure 4.2.

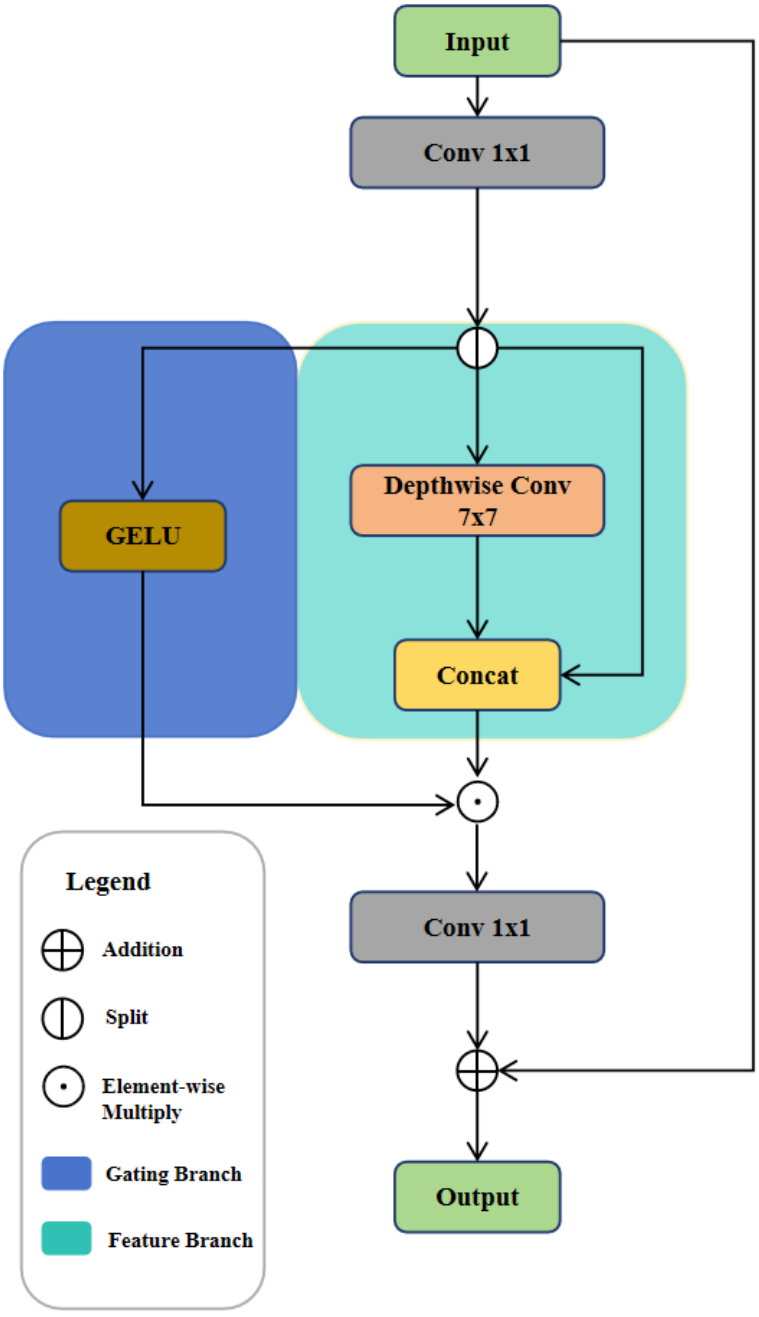


Figure 4.2: Structure of Gated CNN.

Gating Branch: Corresponds to the $\sigma(\text{Proj}_{in1}(X))$ part in the formula. The input feature is first processed through linear projection, then a soft mask with the same spatial dimensions as the feature map is generated through an activation function.

In the dense prediction task of YOLO, UAV aerial images typically contain sub-

stantial high-frequency background noise (such as tree textures and wave patterns), which easily leads to false positives in the detection head. This branch essentially functions as a spatial attention filter. Through training, the model learns to dynamically assign higher weights to potential object regions, while suppressing large background areas. This is equivalent to performing "signal denoising" before features enter the detection head, which significantly improves the signal-to-noise ratio (SNR) of the feature map and helps the YOLO detection head focus more on foreground objects.

Feature Branch: Corresponds to the $\text{DepthConv}_{7\times7}(\text{Proj}_{in2}(X))$ part in the formula. We introduced a 7×7 large kernel convolution for depthwise separable convolution. This design has decisive advantages over the traditional 3×3 convolution in the YOLO backbone. Tiny objects in UAV imagery often have very few pixels, lack intrinsic texture features, and heavily rely on the contextual information from surrounding environments for discrimination. Traditional 3×3 convolution has a limited field of view, while standard deep detectors suffer from excessively large receptive fields that introduce global background noise. Our 7×7 convolution strikes a balance. It expands the local perception scope compared to 3×3 kernels to capture necessary spatial dependencies in shallow layers, yet maintains a compact and focused effective receptive field (ERF) compared to the baselines. This allows the model to perceive small objects and their surrounding environment as an integrated whole, thereby improving detection recall.

Finally, the two branches are fused through element-wise multiplication ($\odot$). This step achieves a dynamic combination of "visual content" and "spatial attention". This design enables our FemtoDet-P2 to maintain the advantages of efficient parallel computation in CNNs while possessing content-adaptive processing capabilities similar to Transformers, thereby significantly enhancing the feature capture ability for tiny objects in complex scenes without introducing additional computational overhead.

4.4 High-Resolution Feature Fusion Neck

Although the use of high-resolution features is not a new concept in object detection literature, standard real-time detectors (such as YOLOv5 to YOLOv10) usually abandon the P_2 layer. This design choice is because in most tasks, small object recall is sacrificed in exchange for lower inference latency, and this loss of precision is acceptable. However, it creates a key "blind spot", in air surveillance, objects smaller than 8×8 pixels will be compressed into sub-pixel representation, or become noise in the P_3 feature map, making subsequent upsampling operations unrecoverable. To solve this problem, we extend the Path Aggregation Network (PANet) to the P_2 layer with a stride of 4, and build a high-resolution feature fusion path. According to the model scaling theory proposed by Scaled-YOLOv4 [47], the optimal network structure should adjust the resource allocation of resolution, depth and width according to the characteristics of input data (such as objects scale distribution). In UAV images, objects are mainly distributed in the area below 16×16 pixels, which means there is a serious waste of resources in the large receptive field and high semantic abstraction of the deep network (P_4, P_5).

Therefore, in FemtoDet-P_2, we added the P_2 layer to solve the semantic gap problem that hinders small object detection. The shallow backbone feature C_2 itself has high-precision object edge information required for localization, but lacks semantic context, leading to False Positives (e.g., misclassifying roof vents as vehicles). On the contrary, the deeper P_3 features carry strong semantic signals but lose spatial information. Crucially, to alleviate the computational overhead that caused P_2 to be abandoned in earlier architectures, we combine this high-resolution "neck" with the efficient MambaOut backbone network. This overall "efficient backbone + heavy neck" design allows us to leverage P_2 without incurring unacceptable inference latency in earlier architectures. Finally, the fused features $\{P_2, P_3, P_4, P_5\}$ are input into four independent decoupled detection heads. The introduction of the P_2 detection head is

specifically designed for objects with area less than 32×32 pixels, which dominate in UAV datasets. Each detector independently predicts object class probability and bounding box coordinates, ensuring that gradient flow from small object optimization does not interfere with the learning of large objects.

4.5 Analysis of Effective Receptive Field

In order to verify the effectiveness of the Gated CNN and P2 layer design, we visualized and quantified the Effective Receptive Fields (ERFs) of the two models. Following the method in [48], we measured the proportion of high-contribution areas under different thresholds (τ) to determine the spatial extent of the input image contributing to the central feature response.

Table 4.1 compares the receptive fields between the baseline model YOLOv10n and our proposed FemtoDet-P2 model. The results show that the most significant difference between the two models occurs near the global threshold, where the ERF size is reduced from 631 pixels (baseline model) to 537 pixels (FemtoDet-P2 model). This 15% reduction indicates that the FemtoDet-P2 model successfully reduces the influence of background noise at long distances. Under the standard ERF threshold, our model maintains a more compact field of view (183 pixels) than the baseline model (211 pixels). This more compact ERF confirms that the high resolution P2 layer limits the over-expansion of the receptive field. In the core area with the highest degree of activation, the receptive field remains concentrated (91 pixels vs. 95 pixels). These indicators show that although the Gated Convolutional Neural Network (Gated CNN) module ensures sufficient local context information capture through its 7×7 convolution kernel, the overall architecture avoids the common "feature dilution" problem in deep detectors. By limiting the receptive field to a more relevant local range, FemtoDet-P2 is more suitable for distinguishing small objects from complex backgrounds.

Table 4.1: Quantitative comparison of Effective Receptive Field (ERF) side lengths (in pixels) between YOLOv10n and FemtoDet-P2 at different contribution thresholds. The input image size is 640×640.

Threshold (τ)	**YOLOv10n**	**FemtoDet-P2**	**Analysis**
$\tau = 0.2$ (Core Focus)	95 px	91 px	Highly Concentrated
$\tau = 0.3$	135 px	121 px	Sharper Boundaries
$\tau = 0.5$ (Effective ERF)	211 px	183 px	**More Compact**
$\tau = 0.99$ (Global Scope)	631 px	537 px	**Noise Suppressed**

4.6 Analysis of Layer-wise Efficacy

In order to further verify the necessity of introducing the high resolution P2 layer into FemtoDet-P2, we conducted a statistical analysis of the detection distribution on different prediction heads. As shown in the Table 4.2, the results on VisDrone2019 and UAVDT datasets show a significant reversal of feature utilization. On the VisDrone2019 dataset, the new P2 layer contributed up to 63.07% of the total detections (18173 targets), exceeding the total contribution of all other layers (P3, P4 and P5). A similar trend was observed on the UAVDT dataset, with P2 contributing 52.57% of the detection amount. This empirically confirms our hypothesis that for UAV aerial images, most of the effective target information is retained in the shallow high-resolution feature map. In contrast, we discover the deep P5 layer shows obvious redundancy. Although the P5 layer has the largest receptive field and the highest semantic abstraction ability (the average confidence on UAVDT is 0.81), its detection contribution rate in UAVDT is only 1.81%, and in VisDrone2019 is only 5.50%. This forms a sharp contrast, indicating that P5 layer consumes a large amount of computing resources in the model, but yields diminishing marginal returns in detection recall.

Table 4.2: Prediction results on VisDrone2019 and UAVDT datasets using the FemtoDet-P2 model. The table details detection counts, proportion, and average confidence for each detection head.

Dataset	Head	Detection Counts	Proportion (%)	Avg Confidence
VisDrone2019	P2	18,173	63.07	0.53
	P3	4,614	16.01	0.46
	P4	4,442	15.42	0.55
	P5	1,585	5.50	0.68
UAVDT	P2	132,674	52.57	0.52
	P3	79,966	31.66	0.54
	P4	35,160	13.93	0.73
	P5	4,558	1.81	0.81

4.7 Summary

In this chapter, we utilize MambaOut as the backbone to enhance the model design for early feature extraction. Although the addition of the P2 layer improves detection accuracy, retaining the P5 layer, which has a large computational cost but a small contribution, brings unnecessary overhead to edge deployment. The results indicate that traditional deeper layers of the feature pyramid are not suitable for UAV aerial imagery where the targets are mainly small. This observation motivates the research presented in the next chapter. In Chapter 5, we will introduce LSCNet, a lightweight architecture that removes redundant deep pyramid layers. By completely shifting the computational focus to shallow feature cascading, it can achieve a more optimized balance, significantly reducing model complexity and inference latency while maintaining the high accuracy of FemtoDet-P2.

Chapter 5

LSCNet: A Lightweight Shallow Feature Cascade Network

5.1 Introduction

In Chapter 4, we introduced the Mambaout backbone network to enhance early feature extraction and optimized it for small targets. However, its deep pyramid structure leads to significant resource waste and the stacked Gated CNN blocks significantly increased the model parameters (from 2.3M to 8.8M) placing a heavy computational burden on the UAV's onboard equipment. Therefore, we need to design a lightweight model. Although YOLOv10 [13] reduces inference latency by eliminating the non-maximum suppression (NMS) post-processing required by traditional YOLO, makes it an ideal choice for resource-constrained environments. However, YOLOv10 is mainly designed for natural images (such as the COCO dataset [15]), where objects are clear and located in the center of the image. Our empirical analysis shows that the model wastes a large amount of computational resources on the deep pyramid layer (P5) to extract high-level semantic abstractions, which are largely redundant for tiny, pixel-limited objects typical in drone perspectives. Based on this finding, we propose LSCNet (Lightweight Shallow Feature Cascade Network), a lightweight architecture

for UAV-based small object detection built upon YOLOv10. Rather than relying on ineffective deep pyramid layers, our approach strategically concentrates on shallow-stage features where small object information is better preserved. By reallocating computational resources from deep pyramid elaboration to shallow feature refinement, LSCNet achieves superior detection performance on both the VisDrone2019 and UAVDT datasets with a compact parameter count.

The design of LSCNet is motivated by the unique challenges of UAV-based object detection. As mentioned earlier, edge deployment on UAV platforms imposes strict constraints on model complexity, power consumption, and memory footprint, thus requiring a lightweight architecture capable of operating in resource-constrained environments. Besides the limitations of the drone platform itself, high-altitude dynamic aerial photography often encounters many adverse factors, such as motion blur, environmental interference, and changes in lighting conditions. In addition, the high-altitude perspective also brings complex background clutter and a greater sense of scale variations [49].

Three areas comprise our primary contributions:

- LoGStem Module [50]: When processing degraded aerial data with inherent blur and noise, conventional object detection backbones frequently struggle with inadequate feature extraction capacity. We use the LoG-Stem module from LEGNet, which was initially created to improve feature representation in low-quality aerial imagery conditions, to overcome this restriction at the network's earliest stage. The LoG-Stem module serves as a robust initial feature extractor that replaces the traditional P1 and P2 layers in YOLO architecture.

- SAOK Fusion Module: To effectively aggregate multi-scale information from the shallow feature cascade layers where small object features are predominantly preserved, we propose the Small-target Aware Omni-Kernel (SAOK) module. Drawing inspiration from the Omni-Kernel Network's multi-branch architecture

for image restoration [51], SAOK strategically adapts the receptive field configuration and attention mechanism specifically for aerial small object detection requirements.

- DyHead [43]: To compensate for potential information loss resulting from the simplified two-layer pyramid architecture and to maximize detection capability within this streamlined framework, we integrate the Dynamic Head (DyHead) as our detection module. Traditional detection heads process features at different scales, spatial locations, and task objectives independently, leading to suboptimal feature utilization. DyHead addresses this limitation through a unified attention framework that coherently optimizes feature representations across three dimensions: scale, space, and task.

5.2 Method

We introduce LSCNet, a novel architecture created specifically for UAV image small object detection. Our approach completely rethinks the YOLO framework by incorporating three key innovations: improved feature extraction, dynamic detection capabilities, and architectural simplification. Our fundamental design idea is optimizing detection accuracy while maintaining model efficiency. Specifically, LSCNet employs a simpler two-layer detection architecture with only two layers with 8× and 16× downsampling detection heads to prevent the loss of small object information caused by deep feature propagation.

In this section, we will explain the detailed implementation of LSCNet; it is structured as follows. First, in Section 5.2.1, we present the deep feature layer efficiency analysis that motivates our Shallow Feature Cascade architecture. Section 5.2.2 introduces the Small-target-aware Omni-Kernel (SAOK) module designed for effective multi-scale feature fusion. Section 5.2.3 describes the LoGStem module utilized for

robust initial feature extraction. Finally, Section 5.2.4 elaborates on the integration of the Dynamic Head (DyHead) mechanism to enhance detection capability through unified attention.

LSCNet's architectural design, illustrated in Figure 5.1, addresses key challenges in UAV-based small object detection through specialized modules. The LoGStem component enhances the initial feature extraction stage by outputting quarter-resolution representations while simultaneously filtering noise through Laplacian-of-Gaussian operations. To enable seamless multi-scale fusion, backbone P4 features are spatially downsampled via PixelUnshuffle, ensuring dimensional alignment with P3-level SPPF outputs after upsampling.

LSCNet adopts the principle of computational efficiency through targeted elimination of resource-intensive deep detection heads. Our SAOK module, combined with architectural adaptations, proves that removing one detection head does not compromise small object detection accuracy. The module processes fused P3–P4 features using multi-scale kernels with ECA attention for enhanced feature refinement. DyHead integration compensates for the reduced head count, delivering improved detection while keeping computational costs acceptable.

5.2.1 Shallow Feature Cascade

Standard detectors typically allocate substantial computational resources to deep pyramid levels (e.g., P5) under the assumption that larger receptive fields universally benefit detection. However, this paradigm, while effective for natural scene images, warrants re-examination in the context of UAV-based small object detection.

To investigate the actual contribution of each pyramid level in aerial small object detection, we conducted a comprehensive quantitative analysis using the standard YOLOv10 architecture on two representative UAV datasets: VisDrone2019 and UAVDT. As detailed in Table 5.1, we evaluated 548 validation images from Vis-

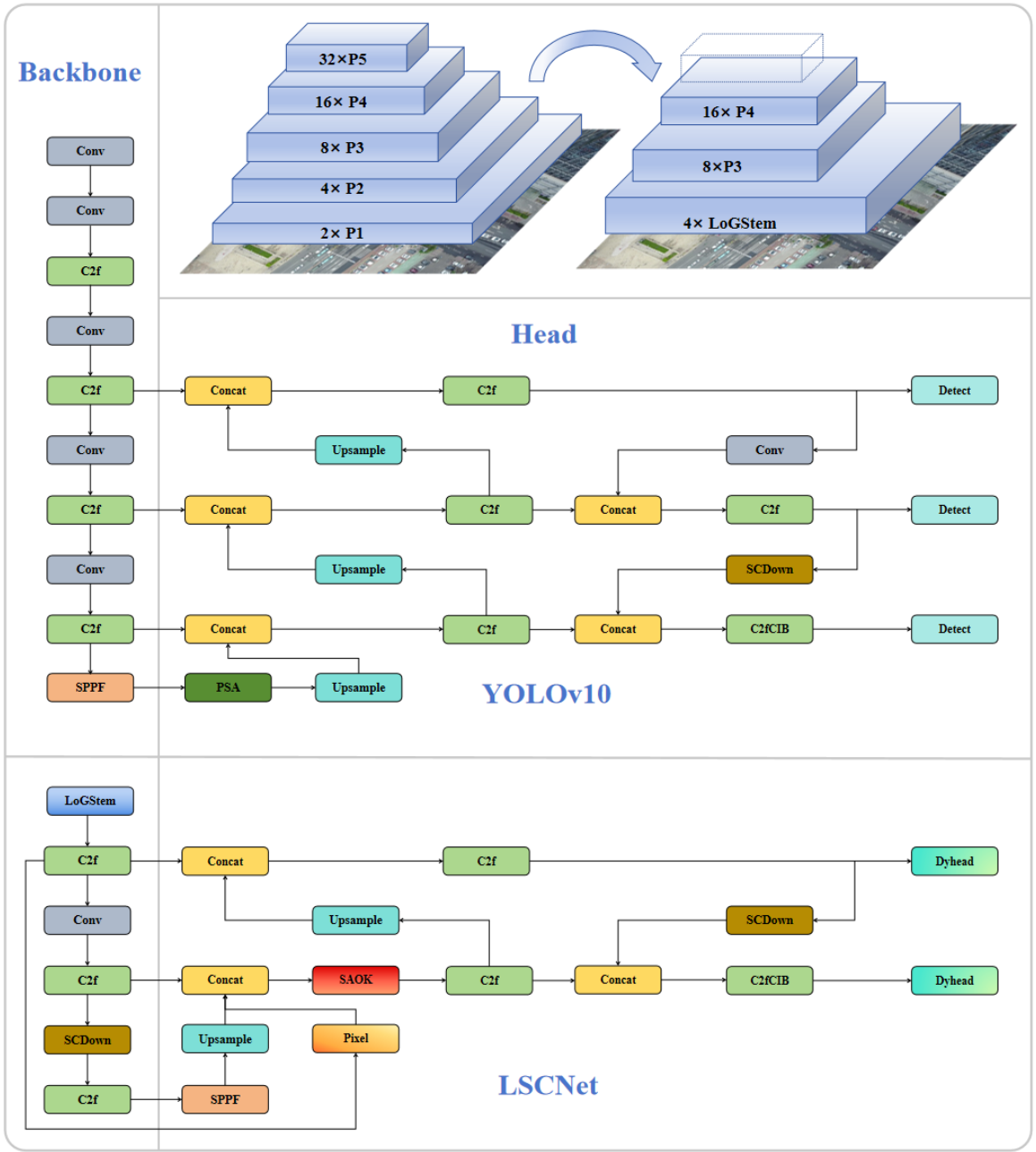


Figure 5.1: Architectural comparison between the baseline YOLOv10 and the proposed LSCNet. The top illustrates the conceptual feature pyramid shift. The middle shows the standard YOLOv10 architecture with three detection heads. The bottom presents the proposed LSCNet architecture.

Drone2019 and analyzed the detection distribution across three pyramid levels (P3, P4, and P5). For VisDrone2019, the P3 layer accounts for 75.92% of all effective detections, while the P4 layer contributes 17.31%. Remarkably, the deepest detection

layer, the P5 layer, produces merely 6.76% of detections despite consuming substantial computational resources. This pattern intensifies in the UAVDT dataset, where P5 contributes less than 3% while P3 and P4 collectively account for over 97% of successful detections. However the bounding box confidence output by the P5 layer is the highest in Visdrone2019 but lower in UAVDT. In the task of small object detection, the overall results were not satisfactory.

Table 5.1: Prediction results on VisDrone2019 and UAVDT datasets using the baseline YOLOv10 model. The table details detection counts, proportion, and average confidence for each detection head.

Dataset	**Head**	**Detection Counts**	**Proportion (%)**	**Avg Confidence**
VisDrone2019	P3	16,182	75.92	0.54
	P4	3,690	17.31	0.56
	P5	1,440	6.76	0.70
UAVDT	P3	155,212	80.02	0.55
	P4	34,056	17.56	0.65
	P5	4,708	2.43	0.29

The basic cause of this phenomenon is that, despite the fact that deep features have larger receptive fields and richer semantic information, feature information of small targets is severely attenuated or even completely disappears during multiple downsampling processes due to the significant reduction in spatial resolution. Shallow features, on the other hand, retain more local details and spatial details.

Motivated by this discovery, we propose the LSCNet architecture, which focuses computational resources on the fusion and augmentation of shallow features rather than the three detection heads of the conventional YOLO structure. In particular, LSCNet only keeps two detection layers. More crucially, this architecture allows us to use more potent initial backbone networks while also drastically reducing the network's computational complexity and parameter count.

Comprehensive experimental validation across both benchmark datasets substantiates the efficacy of our proposed architecture. On the VisDrone2019 dataset, LSC-

Net achieves 44.6% mAP_{50}, representing a substantial improvement of 10.1% over the baseline YOLOv10n. Similarly, on the UAVDT dataset, LSCNet improved by 5.06%, surpassing YOLOv10n's 31.04% mAP_{50}, while reducing the parameter count by 33%.

5.2.2 SAOK

We implement the Small-target-aware Omni-Kernel (SAOK) module during the P3 and P4 feature fusion stage to improve multi-scale feature performance. Inspired by the Omni-Kernel Network [51] in image restoration, the SAOK module uses an adaptive aggregation mechanism of multi-scale convolution kernels to efficiently capture feature information at various scales and improve feature discriminability.

To handle the particular difficulties of UAV aerial scenarios, we do, however, make important adjustments. The SAOK (Small-target-aware Omni-Kernel) module, a key part of our suggested LSCNet architecture, is especially made to handle the feature fusion issues at the crucial P3 layer, where small object information is most noticeable in UAV pictures. SAOK acts as a link between deeper semantic representations and shallow edge characteristics that LoGStem extracts within the LSCNet architecture. SAOK, which comes after the P3 feature extraction stage (see Figure 5.2), combines pixel-unshuffle-processed shallow features with multi-scale contextual information from both P3 and upsampled P4 features. The SAOK module may function at a relatively shallow feature resolution due to this strategic placement, which preserves sufficient spatial representation while gaining rich semantic context for small objects.

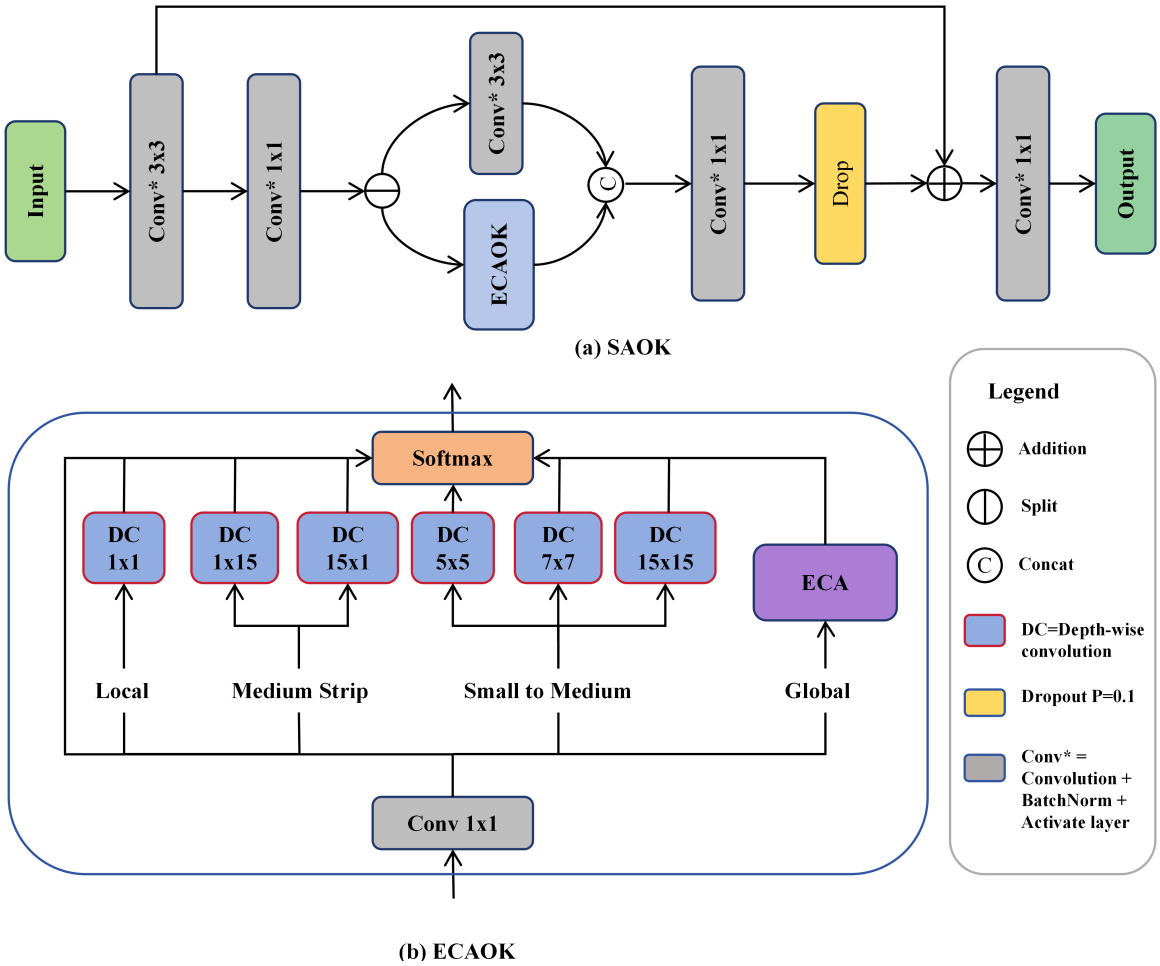


Figure 5.2: Structure of SAOK.

Kernel Configuration for Small Objects: Drawing inspiration from Cui et al.'s Omni-Kernel architecture [51] for image restoration, we adapt the multi-scale convolution strategy specifically for aerial small object detection. Unlike the original design employing ultra-large 63 × 63 kernels optimized for high-resolution image reconstruction, our SAOK adopts a refined kernel configuration tailored to UAV imagery characteristics: 1 × 1 kernels capture local fine-grained details, 5 × 5 and 7 × 7 kernels extract immediate contextual information, and 15 × 15 kernels capture broader spatial relationships. This design philosophy stems from a critical observation: small objects in UAV imagery (typically spanning 10–50 pixels) demand moderate receptive fields that strike a balance between preserving fine-grained features and incorporating sufficient contextual awareness. Additionally, the incorporation of strip-shaped

kernels (1×15 and 15×1) strengthens the module's capability to capture elongated structural patterns—such as vehicle edges and road boundaries—that frequently appear in aerial perspectives.

ECA-enhanced Omni-Kernel (ECAOK): To effectively aggregate the multi-scale features extracted by diverse kernel configurations, we design the ECA-enhanced Omni-Kernel (ECAOK) module as the core computational unit of SAOK. The ECAOK module integrates channel attention mechanism with adaptive multi-scale feature fusion to enhance feature discriminability for small objects. Given an input feature $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, the ECAOK module first applies a lightweight feature transformation:

$$\mathbf{X}_{in} = \mathrm{GELU}(\mathrm{Conv}_{1\times1}(\mathbf{X})) \tag{5.1}$$

where GELU [52] is a nonlinear activation function based on a Gaussian distribution; the formula is

$$\mathrm{GELU}(x) = x \cdot P(X \leq x) = x \cdot \Phi(x) \tag{5.2}$$

It combines the non-saturating property of ReLU with the smoothness of Sigmoid/Tanh, so GELU can be viewed as a smooth variant between ReLU and Sigmoid/Tanh activation functions.

Subsequently, the transformed features are processed through two parallel branches: an Efficient Channel Attention (ECA) branch [42] and a multi-scale depthwise convolution branch. In order to highlight informative channels for small object representation, the ECA branch adaptively recalibrates channel-wise feature responses:

$$\mathbf{X}_{eca} = \mathrm{ECA}(\mathbf{X}_{in}) \tag{5.3}$$

Simultaneously, the multi-scale branch aggregates spatial features across different

receptive fields through weighted depthwise convolutions:

$$\mathbf{F}_{ms} = \sum_{i=0}^{5} w_i \cdot \mathcal{DW}_i(\mathbf{X}_{in}) \tag{5.4}$$

where $\mathcal{DW}_i$ represents the depthwise convolution with kernel sizes $\{1 \times 15,\ 15 \times 1,\ 15 \times 15,\ 1 \times 1,\ 5 \times 5,\ 7 \times 7\}$ and w_i denotes the learnable aggregation weights normalized by the softmax. This adaptive weighting mechanism allows the network to automatically adjust the contribution of each scale according to the characteristics of the input features.

Finally, the outputs from both branches are fused with the original input through a residual connection, followed by activation and projection:

$$\mathbf{X}_{ECAOK} = \mathrm{Conv}_{1\times1}(\mathrm{ReLU}(\mathbf{X} + \mathbf{F}_{ms} + \mathbf{X}_{eca})) \tag{5.5}$$

This design enables the ECAOK module to not only process rich multi-scale spatial information but also effectively discriminate features across different channels. The residual connection further facilitates gradient flow and preserves original details to better serve the recognition of small object features.

5.2.3 LoGStem

Feature extractors are usually inadequate for extracting features from small targets or blurry images for traditional YOLO models. Early on in the image input process, we need a more reliable feature extractor made especially for blurry small targets in complicated situations. We use the LEGNet network's LoG-Stem [50]. The first feature extraction module of the LEGNet network, the LoGStem layer, was created especially to handle noise and edge information loss problems in low-quality remote sensing images. It makes use of the dual capabilities of LoG filters for edge detection and noise suppression, improving edge features early on. Additionally, this feature

extractor's output is downsampled to 1/4 resolution, which precisely matches the purpose of YOLOv10's P1 and P2 layers. As a result, this module serves as the first component of our enhanced network backbone.

The benefits of Laplacian operators and Gaussian smoothing are combined in LoG filters. Equation (5.6) displays the formula for the LoG Kernel. Equation (5.7) displays the formula for the Gaussian Kernel with $\sigma = 1.0$ and a 7×7 kernel size.

$$\mathrm{LoG}_{\sigma}^{k\times k}(\mathbf{x}) = \frac{1}{\pi\sigma^4}\left(1 - \frac{i^2+j^2}{\sigma^2}\right) e^{-\frac{i^2+j^2}{2\sigma^2}} \tag{5.6}$$

$$G_{\sigma}^{k\times k}(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \tag{5.7}$$

The LoGStem structure is shown in Figure 5.3.

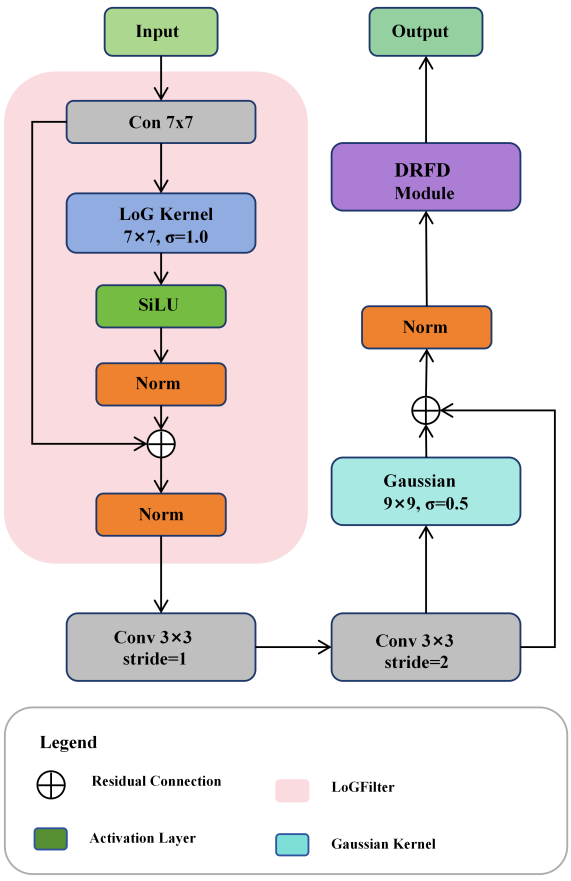


Figure 5.3: Structure of LoGStem module.

5.2.4 DyHead

To compensate for potential performance degradation from the simplified two-layer detection architecture, we integrate the Dynamic Head (DyHead) [43] to enhance the detection capability of our LSCNet. DyHead provides a unified detection head framework through attention mechanisms, coherently integrating three sequential self-attention mechanisms across the three fundamental dimensions of the feature tensor: level, space, and channel.

Scale-aware attention dynamically fuses features from different pyramid levels based on their semantic importance. Complementing our P3- and P4-focused architecture in LSCNet, this attention mechanism adaptively aggregates multi-resolution features to enhance the model's perception capability for objects at varying scales, particularly small objects.

Spatial-aware attention models long-range spatial dependencies through deformable convolution. By concentrating on discriminative regions that consistently coexist at both spatial locations and feature levels, this method strengthens the spatial structure of objects and enhances localization performance in difficult situations like occlusion and motion blur.

Operating on the channel dimension, task-aware attention dynamically switches feature channels ON and OFF to favor distinct detection tasks (classification versus localization). For small objects in UAV imagery that typically exhibit weak and ambiguous feature responses, this mechanism ensures task-specific features—whether for object classification or bounding box regression—are appropriately emphasized during forward propagation. The ablation studies in Section 6.8 validate the effectiveness of this integrated dynamic head design.

5.3 Summary

This chapter introduces LSCNet, a specialized architecture designed for UAV-based small object detection. Based on our empirical analysis, we identified that deep feature layers are redundant for small objects. Therefore, we proposed a shallow feature cascade strategy to shift the computational focus from the P5 layer to the P3 and P4 layers, which contain richer spatial information. This streamlined framework is supported by three core innovations: the LoGStem module, the SAOK module, and the DyHead module. These components work together to enable LSCNet to achieve a balance between inference efficiency and high-precision detection. Chapter 6 presents comprehensive experimental results and comparisons on the VisDrone2019 and UAVDT benchmark datasets, benchmarking our model against state-of-the-art lightweight models.

Chapter 6

Evaluation

6.1 Implementation Details

All experimental procedures were conducted on a computational platform equipped with an Intel Xeon Platinum 8352V processor, 90GB system memory, and an NVIDIA GeForce RTX 4090 graphics card, operating under Ubuntu 20.04.4 LTS. The software stack comprised Python 3.10.14, PyTorch 2.2.2 with CUDA 12.1 support, and the Ultralytics 8.2.50 framework. To ensure equitable comparison across all evaluated architectures, we adopted a consistent training protocol.

All models underwent training from random initialization without leveraging any pre-trained weights, thereby eliminating potential biases from transfer learning. The training regimen consisted of 300 epochs with input images resized to 640 $\times$ 640 pixels and a batch size of 8 samples per iteration. Additional hyperparameters, encompassing optimizer selection and learning rate scheduling mechanisms, adhered to YOLOv8's and YOLOv10's default configuration to preserve experimental reproducibility and maintain consistency across comparative studies.

6.2 Evaluation Metrics

To evaluate the effectiveness of our proposed algorithm on UAV aerial imagery, we employ a combination of accuracy metrics and model complexity indicators. For detection accuracy, we utilize *Precision* (P), *Recall* (R), and Mean Average Precision (mAP) as our primary evaluation metrics. Additionally, we assess the model's computational efficiency through the number of parameters, which directly reflects the model size and deployment feasibility on resource-constrained UAV platforms. The metrics mAP_{50} and mAP_{95} denote the mean AP across all classes at Intersection over Union (IoU) thresholds of 0.5 and 0.95, respectively. IoU represents the overlap ratio between predicted and ground truth bounding boxes, calculated as the ratio of their intersection to their union

In the confusion matrix, True Positives (TP) represent correctly detected objects where both the predicted bounding box and ground truth are positive; False Negatives (FN) occur when actual objects are missed by the detector; False Positives (FP) indicate erroneous detections where the model predicts an object that does not exist; and True Negatives (TN) represent correctly rejected regions.

Based on these foundational concepts, precision measures the proportion of correctly detected objects among all detections made by the model. It reflects the model's ability to avoid false detection and is calculated as

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

Recall evaluates the proportion of actual objects that are successfully detected by the model. It indicates the model's capability to identify all existing objects and is defined as

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

To balance false positives and false negatives, therefore considering both *Precision*

and *Recall*, the *F1* score is the harmonic mean of *Precision* and *Recall*. This prevents models from achieving high scores by sacrificing one metric. The formula is

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6.3}$$

IoU is calculated as the ratio of the intersection area to the union area between a predicted bounding box B_p and its corresponding ground truth B_{gt}, where $B_p \cap B_{gt}$ denotes the intersection area and $B_p \cup B_{gt}$ denotes the union area of the two boxes. The formula is

$$IoU = \frac{Area(B_p \cap B_{gt})}{Area(B_p \cup B_{gt})} \tag{6.4}$$

Average Precision (AP) is then computed as the area under the Precision–Recall curve for a specific object category at a given IoU threshold, where $P(R)$ denotes the precision as a function of recall R and dR represents the differential element of recall used to compute the integral area under the PR curve. The formula is

$$AP = \int_0^1 P(R)dR \tag{6.5}$$

Mean Average Precision (mAP) extends this concept by averaging AP values across all object categories:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{6.6}$$

where N represents the number of object categories and $AP(i)$ denotes the average precision for category i. In our experiments, we report both mAP_{50} (mAP at IoU threshold of 0.5) and mAP_{95} (mAP at IoU threshold of 0.95), which provides a more stringent evaluation of localization accuracy.

GFLOPs (Giga Floating-point Operations) quantifies the theoretical computational complexity of the model. It represents the number of billion floating-point operations required to perform a single forward pass on an input image. The formula

is

$$GFLOPs = \frac{TotalFloatingPointOperations}{10^9} \tag{6.7}$$

Lower *GFLOPs* indicate reduced computational demand. Although the actual performance may vary across different hardware platforms due to optimization differences, GFLOPs generally serves as a general metric for evaluating the computational resources required to run a model.

FPS (Frames Per Second) measures the inference speed, indicating the number of images the model can process in one second. It is calculated as

$$FPS = \frac{N_{frames}}{T_{elapsed}} \tag{6.8}$$

where N_{frames} denotes the total number of processed frames and $T_{elapsed}$ represents the total elapsed time in seconds. Higher FPS indicates faster model inference speed.

6.3 Datasets

We conducted experiments on two challenging UAV datasets, VisDrone2019 and UAVDT. Both datasets contain various scenarios, including different city and weather conditions.

As shown in Figure 6.1, we conducted a statistical analysis of object size distribution in the two datasets. Following the COCO standard [15], objects are categorized into small (area less than 32^2 pixels), medium (area between 32^2 and 96^2 pixels), and large (area greater than 96^2 pixels). The statistics reveal that both datasets are heavily dominated by small objects, which constitute over 60% of the samples in both datasets. Large objects are extremely rare in both datasets, particularly in UAVDT, where they account for less than 3%.

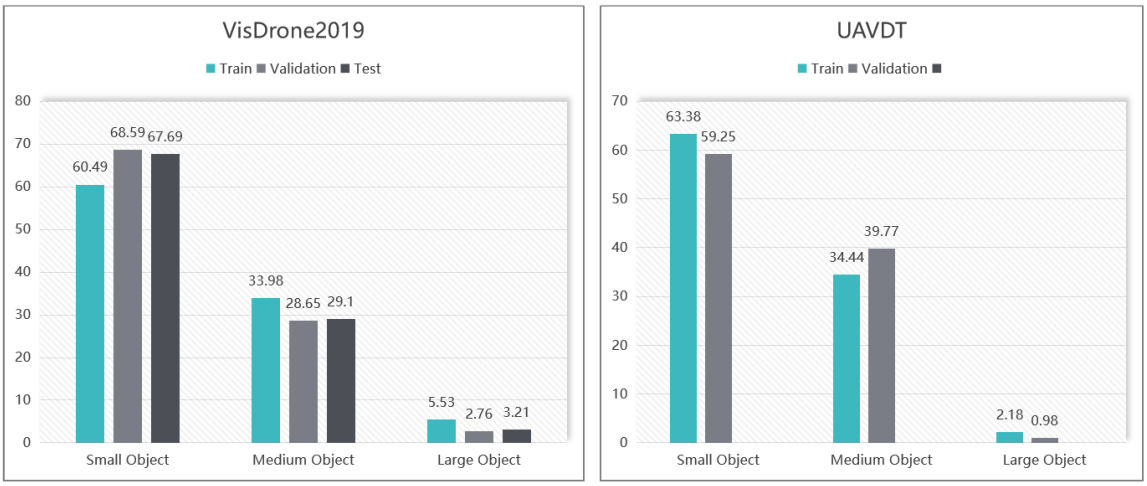


Figure 6.1: Object size distribution across VisDrone2019 and UAVDT datasets, showing predominance of small objects (60–68%) and limited large objects (less than 6%).

6.4 VisDrone

The VisDrone2019 dataset includes 10,209 still photos and 288 video clips with 261,908 frames taken in various cities and situations and was created by the AISKY-EYE team at the Machine Learning and Data Mining Laboratory of Tianjin University. This dataset's object detection task comprises 3190 test photos, 548 validation images, and 6471 training images with annotations for ten object categories, such as trucks, cars, vans, and pedestrians. With a total of 2.6 million annotation boxes, it poses major hurdles typical of real-world drone applications, including occlusions, dense tiny objects, and variable illumination and weather conditions across many places and settings. This dataset is very useful for assessing model performance and proving the efficacy of our suggested approach because it covers a wide range of UAV aerial surveillance scenarios. Additionally, we compare this dataset with a few recently released lightweight models.

6.5 UAVDT

The UAVDT (UAV Detection and Tracking) benchmark was released by the Computer Vision Laboratory at Shenzhen University in 2019. This comprehensive dataset comprises over 10 h of raw videos with approximately 80,000 representative frames extracted from 50 video sequences. The sequences are captured under diverse real-world scenarios, including different weather conditions, varying camera views, different altitudes, and various urban and rural environments. The dataset provides detailed annotations with over 2.8 million bounding boxes across three primary categories: car, truck, and bus. Due to its focus on vehicle detection from UAV perspectives, UAVDT has been widely adopted in the computer vision community as an important benchmark for evaluating detection model performance under realistic UAV operational conditions.

UAVDT is uniquely structured as a dual-purpose benchmark for both object detection and tracking tasks. The motion blur introduced during UAV flight significantly increases detection difficulty, as objects may appear distorted across frames. However, this characteristic makes UAVDT more representative of real-world scenarios where UAVs encounter dynamic conditions such as platform instability and varying flight speeds.

6.6 Ablation Study of FemtoDet-P2

In order to verify the contribution of each component proposed in FemtoDet-P2, we conducted ablation experiments on the VisDrone2019 validation set. We take the standard YOLOv10n as the baseline, and gradually introduce MambaOut backbone network and high-resolution P2 layer. First, we replace the CSP-based backbone network in YOLOv10n with MambaOut architecture (using the gated CNN module), while maintaining the original neck structure (P3-P5). As shown in the second row

of Table 6.1, this improvement has significantly enhanced the model's performance. mAP_{50} increased by 3.6% (from 34.80% to 38.40%), and mAP_{95} increased by 2.47%. Although the model parameters increased from 2.2 million to 8.8 million due to the stacked gated CNN modules, the inference speed (FPS) remained at the high level of 70.4 FPS. This validates our hypothesis in Chapter 4: the gated CNN mechanism can effectively capture global context and suppress background noise, thereby providing better feature representation for UAV images without the hardware latency issues typically associated with SSMs. Based on the MambaOut backbone network, we further introduced the P2 layer to build the complete FemtoDet-P2 model. This step aims to recover the details of small objects lost during the downsampling process. Comparing the second and third rows of Table 6.1, after the P2 layer is introduced, mAP_{50} further increases to 40.35% and mAP_{95} to 24.13%. Notably, the performance improvement is particularly significant for small-scale and complex categories. For example, in the 'Pedestrian' category of Table 6.2, the mAP_{95} improves from 19.3% (MambaOut backbone) to 22.0% (FemtoDet-P2), and in the 'Motor' category, it increases from 19.0% to 22.1%. This demonstrates that high-resolution P2 features are crucial for detecting small objects (less than 16×16 pixels).

The integration of the P2 layer introduces additional computational costs. GFLOPs increase from 22.7 to 28.1, and inference FPS decreases from 70.4 to 60.2. However, this still fully satisfies the real-time requirements of airborne equipment. Although FemtoDet-P2 has 9.2 million parameters (higher than the baseline), its mAP_{50} increases by 5.55%, which demonstrates that our proposed "efficient backbone + deep neck" design strategy effectively detects small objects while maintaining practical efficiency for airborne platforms.

Table 6.1: Ablation study of FemtoDet-P2 on the VisDrone2019 validation set.

Model	**mAP**$_{50}$ (%)	**mAP**$_{95}$ (%)	**Params** (M)	**GFLOPs**	**FPS**
YOLOv10n	34.8	20.0	2.2	6.7	133.7
+ MambaOut Backbone	38.4	22.5	8.8	22.7	70.4
+ P2 Neck (**FemtoDet-P2**)	**40.4**	**24.1**	9.2	28.1	60.2

Table 6.2: Per-class performance comparison (mAP$_{50}$ and mAP$_{95}$).

Class	**mAP**$_{50}$ (%)			**mAP**$_{95}$ (%)		
	Base	**MambaOut**	**Ours**	**Base**	**MambaOut**	**Ours**
Pedestrian	37.5	42.6	**46.8**	16.4	19.3	**22.0**
People	29.5	33.9	**38.2**	11.5	13.3	**15.3**
Bicycle	10.0	13.1	**13.5**	4.2	5.1	**6.0**
Car	76.7	79.1	**82.0**	53.4	55.8	**58.5**
Van	39.3	43.4	**45.4**	26.6	29.9	**32.1**
Truck	29.4	34.3	**34.4**	18.0	21.6	**22.6**
Tricycle	23.5	26.2	**26.5**	12.1	14.3	**15.2**
Awning-tri	13.8	14.6	**14.5**	8.4	8.9	**9.5**
Bus	48.5	53.2	**54.4**	32.8	38.0	**38.1**
Motor	39.7	43.7	**47.9**	17.1	19.0	**22.1**
Average	34.8	38.4	**40.4**	20.0	22.5	**24.1**

6.7 Visualization Analysis of FemtoDet-P2

To intuitively demonstrate the effectiveness of FemtoDet-P2 in feature extraction and background suppression, we used Grad-CAM [53] to visualize the attention heatmaps. In a typical urban aerial scene containing dense small objects, Figure 6.2 shows the baseline model (left) and FemtoDet-P2 (right).

The baseline heatmap exhibits a "feature dispersion" phenomenon. The active area is relatively diffuse, and the intensity of the object (vehicle) is weak (mainly displayed in yellow or light green). The key is that there is serious background interference in the baseline model, and attention is allocated to non-object high-frequency texture regions, such as building windows on the left and trees on the right. This shows that the standard CSP backbone network struggles to filter out the complex environmental noise in UAV imagery.

In contrast, FemtoDet-P2 shows a stronger ability to distinguish objects. The red area is highly concentrated, indicating that the attention is more focused on the object. Even for small vehicles located at the image periphery, our model can generate clear activation peaks, which verifies the contribution of the high-resolution P2 layer to the preservation of spatial details. Background areas, such as roads and buildings, remain dark blue. This proves that the gated CNN module can effectively play the role of a spatial filter, suppress background noise, and enable the model to focus on foreground objects. In conclusion, compared with the baseline model, the targeted design of FemtoDet-P2 achieves a significantly higher signal-to-noise ratio (SNR), which verifies that it can achieve more robust detection of small objects.

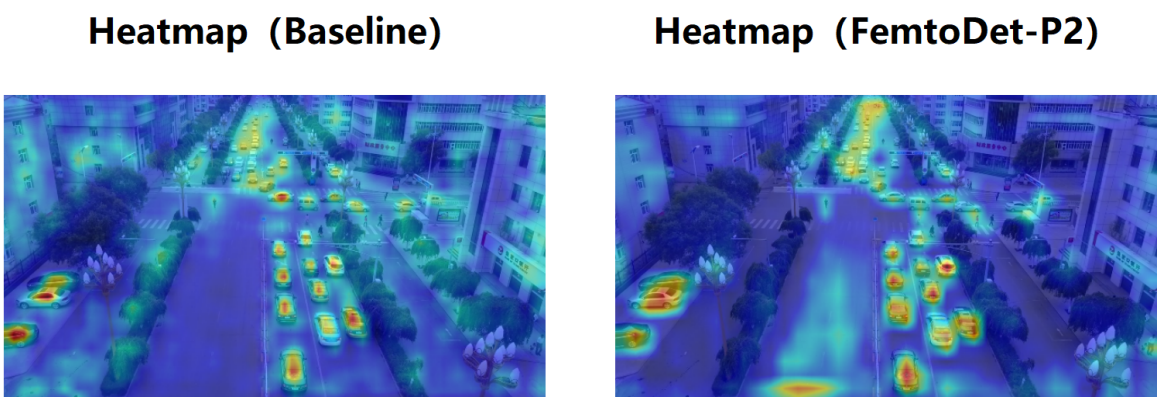


Figure 6.2: Attention heatmap visualization comparing baseline YOLOv10 and FemtoDet-P2 on VisDrone2019 dataset.

6.8 Ablation Study of LSCNet

To validate the effectiveness of each component in LSCNet, we designed ablation experiments with single modules and module combinations. The experiments shows in Table 6.3 use YOLOv10n as the baseline and are evaluated on the VisDrone2019-val dataset. All performance metrics, including mAP_{50}, mAP_{95}, parameter count, and GFLOPs, are obtained through YOLO's official validation script using fused parameters. FPS values are calculated as the average of five independent test runs. Each test is preceded by a 200-image GPU warm-up, followed by measuring the total inference time on 1000 images.

The ablation study is conducted in two parts. First, we independently evaluate the contribution of each module. LoGStem improves mAP_{50} by 2.6 percentage points through enhanced shallow feature extraction; SAOK achieves a 2.1 percentage point improvement by optimizing the feature pyramid structure and integrating spatial-aware mechanisms, while significantly reducing parameters to 0.84 M; and DyHead brings a 1.1 percentage point performance gain through dynamic detection head mechanisms. Notably, SAOK not only significantly reduces parameters but also increases FPS from 133.7 to 160.3, optimizing computational efficiency while improving accuracy.

Second, we also evaluate the synergistic effects of different module combinations. The combination of LoGStem and SAOK achieves 40.7% mAP_{50}, validating the complementarity between shallow feature enhancement and optimized pyramid architecture. When integrating all three modules, the model reaches optimal performance: mAP_{50} of 44.6%, an improvement of 10.1 percentage points over the baseline, and a mAP_{95} improvement of 7.3 percentage points. The complete model uses only 1.48 M parameters, a 32.7% reduction compared to the baseline.

Table 6.3: Ablation study results on VisDrone2019-val dataset. All metrics are evaluated using YOLOv10n as baseline with fused parameters. FPS is averaged over five runs. SAOK represents SAOK module with optimized feature pyramid.

Model	**mAP_{50}** (%)	**mAP_{95}** (%)	**Params** (M)	**GFLOPs**	**FPS**
(a) Single-block ablation study					
Baseline	34.8	20.0	2.2	6.7	133.7
+ LoGStem	37.4	22.0	2.79	16.7	128.3
+ SAOK	36.9	22.3	0.84	11.8	160.3
+ DyHead	35.9	21.0	2.80	7.7	84.2
(b) Combination ablation study					
Baseline	34.8	20.0	2.2	6.7	133.7
+ LoGStem + SAOK	40.7	24.5	0.87	21.5	137.2
+ LoGStem + DyHead	39.9	23.8	3.27	17.9	78.4
+ SAOK + DyHead	41.2	24.7	1.46	17.5	107.4
+ All Modules	44.6	27.2	1.48	27.3	101.2

6.9 Performance of LSCNet on VisDrone2019

LSCNet was created especially for situations involving the detection of small objects. Compared to baseline models, our approach demonstrates outstanding performance on this benchmark while maintaining extremely lightweight characteristics. We chose a number of recent outstanding models from relevant disciplines for comparison, as summarized in Table 6.4.

Comparison with lightweight models: LSCNet demonstrates superior efficiency by achieving 44.6% mAP_{50} with only 1.48 M parameters and 27.3 GFLOPs. Among models with comparable parameter counts, LSCNet exhibits a significant performance advantage: it surpasses OSD-YOLOv10 (1.6M parameters), the model with the most similar size, by 11.2 percentage points in mAP_{50} while using fewer parameters. Furthermore, LSCNet matches the detection performance of LRDS-YOLO with 64.5% fewer parameters, reducing the model size from 4.17 M to 1.48 M parameters.

Comparison with specialized UAV detection models: LSCNet demonstrates strong competitive performance. While achieving a 3.2 percentage point improvement in mAP_{50} over EL-YOLO, our method utilizes 65.5% fewer parameters than EL-YOLO's 4.29M. Although TA-YOLO-S achieves slightly higher accuracy with a 0.8 percentage point lead in mAP_{50}, it requires 9.4 times more parameters (13.9 M) than LSCNet, highlighting our model's substantial advantage in parameter efficiency.

As illustrated in Figure 6.3, we present a three-way comparison featuring Ground Truth (GT) annotations, baseline model outputs, and LSCNet predictions. In the upper example showing an intersection scene, the GT labels 68 objects and the baseline captures 49, while LSCNet identifies 76 objects. The lower example presents a congested urban traffic environment where the GT marks 186 objects, the baseline detects only 72, and LSCNet achieves 113 detections.

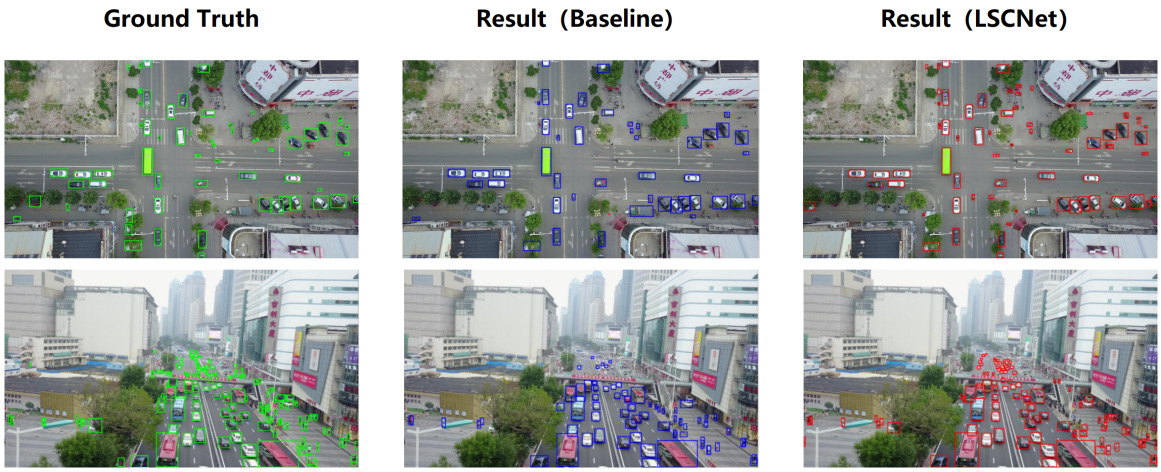


Figure 6.3: Comparison of detection results on VisDrone2019 dataset between baseline model and our proposed LSCNet.

The attention mechanism analysis in Figure 6.4 provides additional insights: the baseline model (left panel) shows a scattered attention distribution with considerable background interference, especially in areas with poor contrast, whereas LSCNet (right panel) exhibits focused attention responses targeting small objects, including roadside pedestrians and faraway vehicles, while effectively minimizing irrelevant background activation.

Through visual analysis of both detection outputs and attention visualization, LSCNet demonstrates enhanced capability in detecting small-scale objects such as pedestrians, bicycles, and distant vehicles, substantially exceeding baseline performance. These improvements confirm the soundness and efficacy of the LSCNet framework.

To further analyze the performance of LSCNet, we provide the normalized confusion matrix for the VisDrone2019 dataset in Figure 6.5. The horizontal axis represents the ground truth labels, while the vertical axis shows predicted labels.

In the matrix, the cells along the main diagonal represent instances where the model's prediction matches the ground truth. High values in these cells indicate robust classification accuracy for the respective categories, also called True Positives. Unlike standard object categories, the "background" column and row represent the

absence of an object. In the column of "background", the model incorrectly predicts background clutter as a specific object, also known as False Positives. In the background row, the model fails to detect an existing object, effectively categorizing it as "background", which are False Negatives.

For instance, the high value of 0.70 in the cell (Background, Pedestrian) indicates a relatively high value of missed detections for small pedestrians. Additionally, notable inter-class confusion exists between visually similar categories. For example, 37% of "vans" were incorrectly classified as "cars", and semantic confusion exists between "pedestrian" and "people." This may be attributed to the low resolution of aerial images, which makes it difficult to discern subtle structural differences between vehicle types, as well as semantic ambiguity in the dataset annotation standards. These are common challenges in high-altitude aerial imagery.

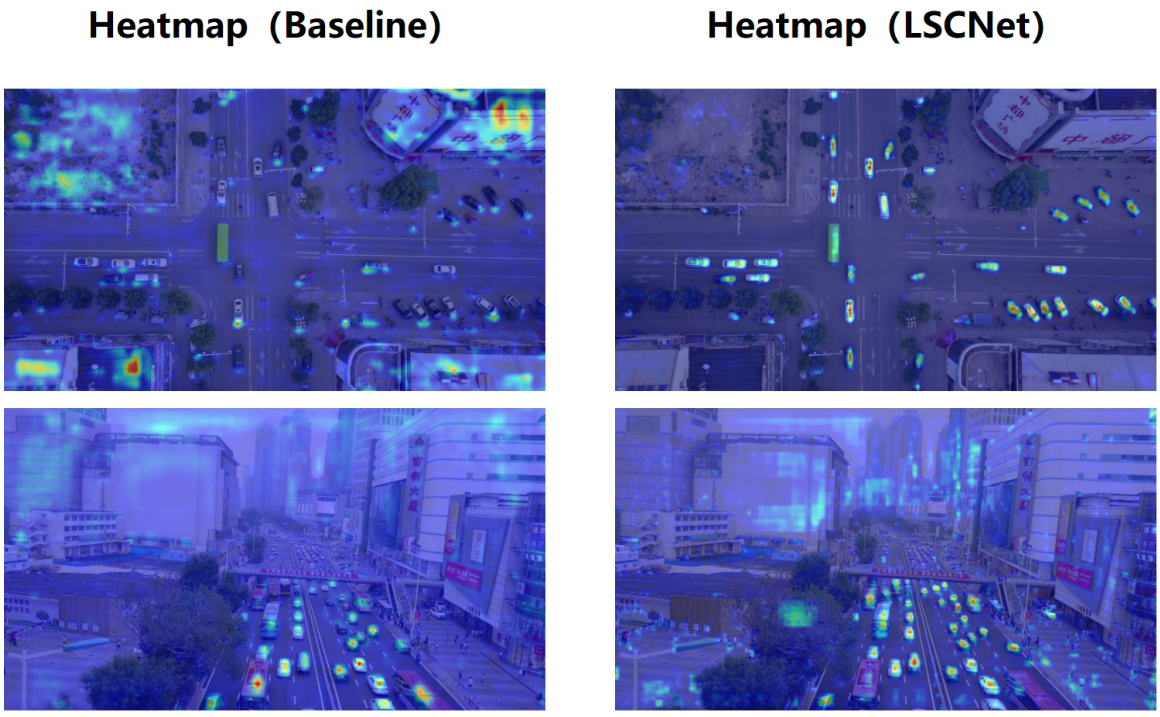
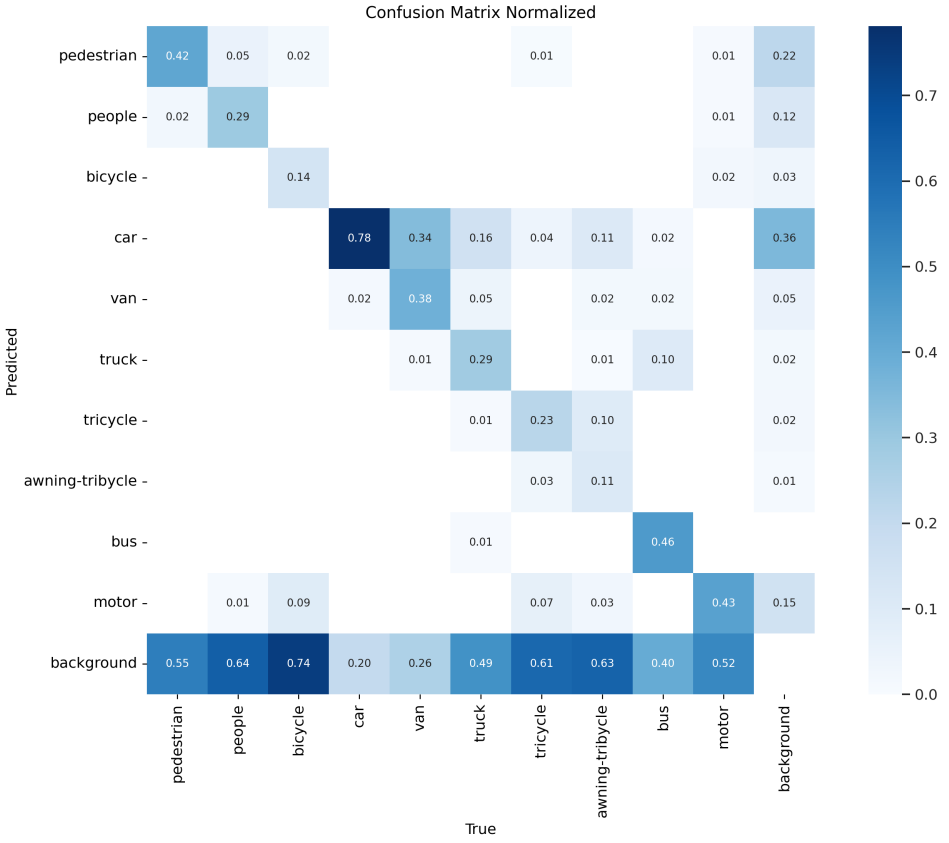
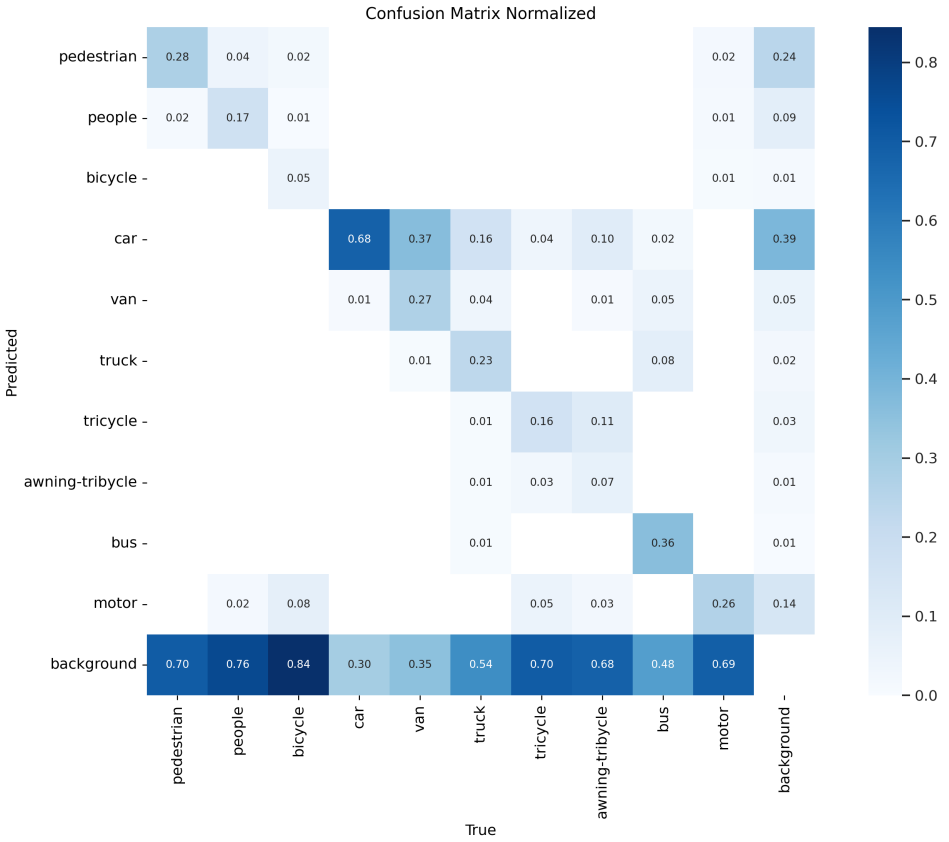


Figure 6.4: Attention heatmap visualization comparing baseline YOLOv10 and LSCNet on VisDrone2019 dataset.



(a) LSCNet



(b) YOLOv10

Figure 6.5: Confusion matrix on VisDrone2019 dataset.

Table 6.4: Comparative experiment results for different models on VisDrone2019 dataset. Dashed lines represent unavailable data. Part of the data is from [2].

Model	**Prec.** (%)	**Recall** (%)	**F1** (%)	**mAP**$_{50}$ (%)	**mAP**$_{95}$ (%)	**Params** (M)	**GFLOPs**
YOLOv3-tiny	39.1	24.3	22.5	23.6	13.2	9.52	14.3
YOLOv5n	44.5	33.2	38.0	32.9	19.1	2.18	5.8
YOLOv5s	51.1	38.1	43.7	39.3	23.4	7.81	18.8
YOLOv5m	47.7	36.8	37.0	39.4	23.0	20.88	48.0
YOLOv5l	50.7	38.6	43.9	41.4	24.6	46.15	107.8
YOLOv5x	52.1	40.4	45.5	41.0	26.0	86.20	203.9
YOLOv6s	40.3	30.5	30.2	30.2	17.7	4.15	11.5
YOLOv7-tiny	47.6	37.3	41.8	35.8	18.8	6.04	13.3
YOLOv8n	45.0	33.0	38.1	33.1	19.2	2.68	6.8
YOLOv8s	50.7	37.9	43.3	39.1	23.4	9.83	23.4
YOLOv8m	53.3	41.1	46.4	42.5	26.0	23.26	67.5
YOLOv9s	52.0	38.0	39.4	43.9	23.8	6.19	22.1
YOLOv10n	45.0	34.5	39.1	34.5	19.9	2.26	6.5
YOLOv10s	52.7	38.0	44.0	39.8	23.8	7.22	21.4
YOLOv10m	55.1	42.1	47.4	44.2	26.9	15.31	58.9
YOLOv11n	42.7	32.7	37.3	32.2	18.6	2.61	6.5
YOLOv11s	49.9	38.7	43.5	39.4	23.6	9.41	21.3
YOLOv11m	55.7	42.5	48.2	44.1	27.2	20.03	67.7
YOLOv11l	55.5	43.0	48.3	44.4	27.5	25.28	86.6
RT-DETR-R18 [11]	57.2	40.0	47.1	41.4	25.1	20.57	60.0
EL-YOLO [16]	48.6	39.0	42.0	40.6	23.6	4.29	24.7
CPDD-YOLOv8 [54]	51.7	41.7	46.1	41.0	23.5	206.0	141.9
OSD-YOLOv10 [17]	43.9	32.5	37.4	33.4	19.1	1.60	7.9
TA-YOLO-S [55]	53.9	44.3	48.6	45.4	27.7	13.90	43.3
LSOD-YOLO [56]	48.4	38.2	42.7	37.0	-	3.80	33.9
LRDS-YOLO [2]	53.3	41.6	46.0	43.6	26.6	4.17	24.1
LSCNet (Ours)	**52.5**	**42.6**	**47.1**	**44.6**	**27.2**	**1.48**	**27.3**

6.10 Performance of LSCNet on UAVDT

Training and validating models on different datasets is crucial for assessing model performance and robustness. It can greatly enhance object identification models' performance, robustness, and capacity for generalization, guaranteeing their efficacy

in a range of real-world applications. We carried out thorough training and testing experiments on the UAVDT dataset to further confirm the LSCNet architecture's universality. The LSCNet model exhibits better detection performance, as indicated in Table 6.5 , with improvements in Precision and mAP_{50} of 14.5% and 5.06%, respectively. LSCNet outperforms YOLOv10n by 3.63% for the mAP_{95} measure in particular, demonstrating consistent detection performance across various IoU thresholds.

We can see that the LSCNet model does better at identifying small objects by examining the accuracy performance across various categories. Compared with YOLOv10n, the mAP_{50} for the truck category increases by 6.58% with a precision improvement of 20.35%; for the bus category, mAP_{50} increases by 13.94%, precision improves by 21.6%, and recall improves by 13.05%. These findings show that LSCNet offers notable architectural benefits for detecting small objects, validating that the synergistic effect of the LoGStem module, SAOK attention mechanism, and DyHead detection head can effectively enhance the model's capability for small object detection across different datasets.

Table 6.5: Performance comparison on UAVDT dataset between YOLOv10n and LSCNet.

Class	**YOLOv10n**				**LSCNet**			
	P	**R**	**mAP$_{50}$**	**mAP$_{95}$**	**P**	**R**	**mAP$_{50}$**	**mAP$_{95}$**
all	32.25	34.10	31.06	18.45	46.75	35.15	36.12	22.08
car	78.25	78.00	83.29	49.47	79.8	69.75	77.93	46.64
truck	9.27	10.38	3.56	1.91	29.62	8.72	10.14	6.99
bus	9.22	13.93	6.34	3.97	30.82	26.98	20.28	12.61

Figure 6.6 compares detection results across three columns: Ground Truth (GT) annotations, baseline model predictions, and our LSCNet predictions. The first row

shows an open highway scene under sufficient lighting conditions, where GT contains 15 annotated objects, while the baseline model detects only 13 objects and our LSCNet successfully identifies 20 objects. This demonstrates that LSCNet's detection capability for vehicles in open high-altitude scenarios with good lighting far surpasses that of the baseline model. Examining the corresponding attention heatmap reveals that the model exhibits extremely high attention to small vehicle objects, while also allocating considerable attention to vehicles parked on both sides of the distant roadway.

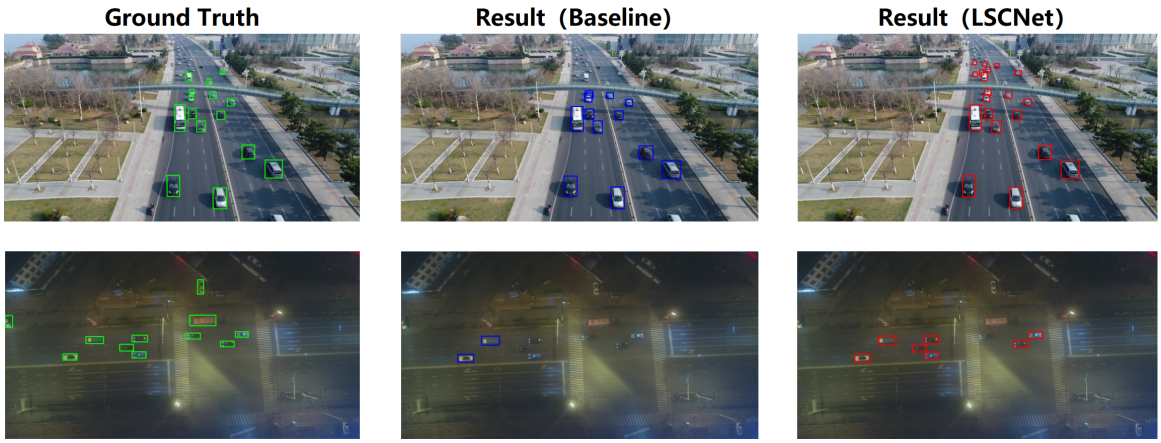


Figure 6.6: Comparison of detection results on UAVDT dataset between baseline model and our proposed LSCNet.

The second row presents detection performance in blurred nighttime imagery, where LSCNet correctly identifies the majority of annotated vehicles on the road, showing substantial improvement over the baseline model. GT contains 11 annotated objects, excluding vehicles in shadowed areas and those parked outside the roadway, and the baseline model detects only 2 objects, whereas our LSCNet successfully identifies 7 objects.

The attention heatmap (Figure 6.7) visualization demonstrates that the model allocates attention to nearly all vehicles in the image. Notably, vehicles parked in the shadowed area at the top of the image, which are barely visible to the human eye, also receive considerable attention from the model, though they were ultimately not in-

cluded due to confidence scores falling below the threshold. These two examples fully demonstrate our proposed model's adaptability to various complex environments.

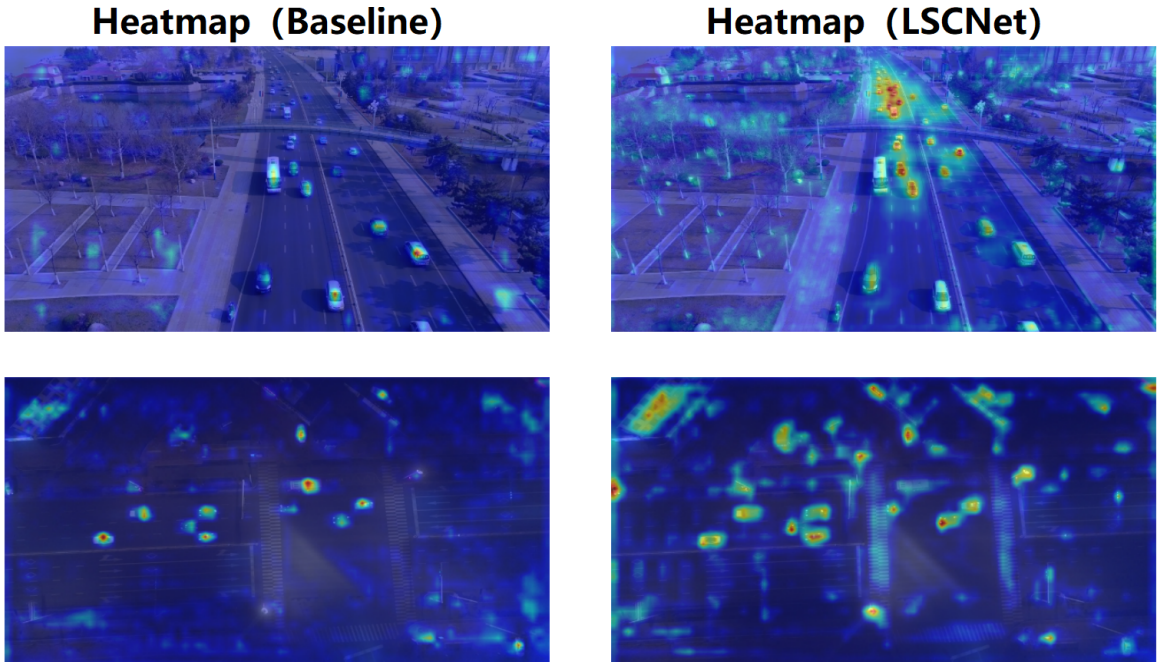


Figure 6.7: Attention heatmap visualization comparing baseline YOLOv10 and LSCNet on UAVDT dataset.

6.11 Discussion of LSCNet

As shown in Table 6.4, we compare LSCNet with numerous recent lightweight models in this domain. The results demonstrate that our method achieves superior performance while maintaining fewer parameters compared to other state-of-the-art models.

When compared to the YOLOv10 family, LSCNet (1.48 M params) outperforms not only the baseline YOLOv10n but also the significantly larger YOLOv10s (7.22 M params) and YOLOv10m (15.31 M params). Specifically, LSCNet achieves a higher mAP_{50} (44.6%) than YOLOv10m (44.2%) while utilizing less than 10% of its parameters. Furthermore, LSCNet surpasses the Transformer-based RT-DETR-R18 [11] (+3.2% mAP_{50}), showing that for edge-based small object detection, a CNN model is currently more efficient than heavy Transformer architectures.

Compared to recent specialized UAV models, LSCNet shows distinct advantages. Specifically, compared with OSD-YOLOv10 [17], which has a similar parameter count (1.6 M), LSCNet leads by a substantial margin of 11.2% in mAP_{50}, demonstrating that our architectural innovations (LoGStem and SAOK) provide far superior feature representation than standard lightweight modifications. When compared with the high-performing TA-YOLO-S [55], which has 45.4% mAP_{50}, LSCNet achieves comparable accuracy within 0.8% difference but with a drastic reduction in model complexity, using only 1/9th of the parameters (1.48 M vs. 13.9 M) and 37% fewer GFLOPs (27.3 vs. 43.3). Furthermore, LSCNet outperforms LRDS-YOLO [2] by 1.0% in mAP_{50} while reducing parameters by 64.5% and maintaining a similar computational cost. To provide a more intuitive comparison, we visualize the performance of lightweight models from Table 6.4 in a radar chart, shown as Figure 6.8.

Despite the promising results, our study has limitations. First, while the parameter count is minimized, the increased GFLOPs due to high-resolution processing slightly reduce the inference speed (FPS) compared to the ultra-fast YOLOv10n. However, when compared with YOLOv10m, which achieves similar accuracy, LSCNet demonstrates a substantial reduction in GFLOPs (27.3 vs. 58.9), representing a 53.6% decrease in computational cost.

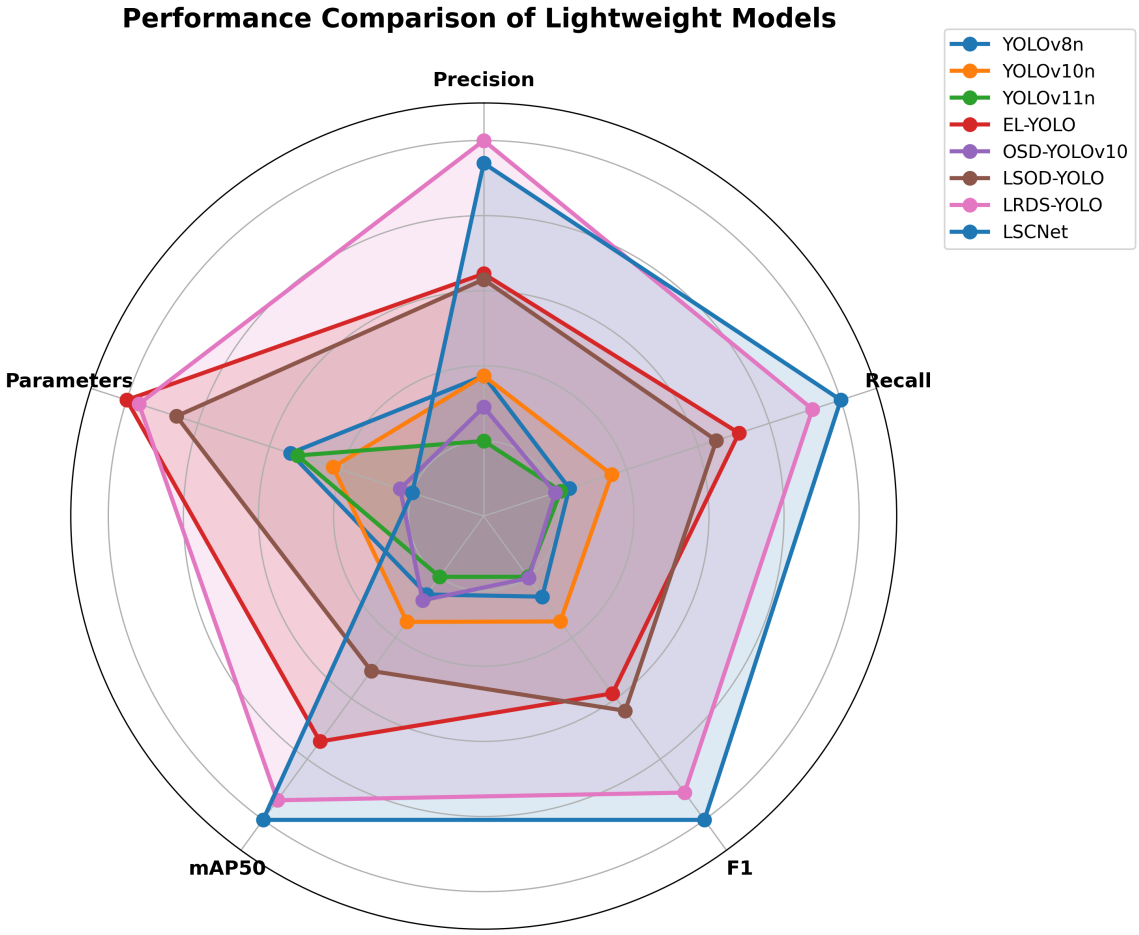


Figure 6.8: Radar chart comparing performance metrics across lightweight models.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis focuses on the key challenges of small object detection in UAV aerial images. We systematically analyzed the limitations of existing real-time detectors, especially the information loss caused by deep downsampling and the redundancy of deep semantic layers. In order to solve these problems, we proposed two new architectures: FemtoDet-P2 and LSCNet, and proved that optimizing feature resolution and focusing on shallow feature enhancement is the key to balancing edge platform detection accuracy and computational efficiency.

First, in Chapter 4, we explored the design strategy of "efficient backbone + deep neck" by introducing MambaOut backbone and P2 architecture. By using Gated CNN mechanism, we successfully suppressed background noise and captured global context information without the SSM hardware latency. We found that the introduction of high-resolution P2 layer can significantly restore the spatial details of small objects (less than 16×16 pixels). FemtoDet-P2 reached 40.4% of mAP_{50} on the VisDrone2019 dataset, verifying that high-resolution features are crucial for aerial object detection, although this method increases the computational cost.

Based on the research results of FemtoDet-P2, we discussed the redundancy of

deep feature layers in Chapter 5. Our empirical analysis shows that in the UAV scenario, the deep P5 layer contributes less than 7% to the detection performance. Therefore, we propose a lightweight architecture LSCNet based on shallow feature cascade strategy. We replaced the inefficient deep pyramid layers with targeted shallow feature enhancement modules: LoGStem module for robust initial edge extraction, SAOK module for P3/P4 multi-scale feature fusion, and DyHead module for unified attention. This design successfully shifted the focus of computation from semantic abstraction to spatial refinement.

Finally, in Chapter 6, we comprehensively evaluated the performance of LSCNet with state-of-the-art lightweight detection models on the VisDrone2019 and UAVDT benchmark datasets. The results show that LSCNet achieves higher detection accuracy with fewer parameters. On the VisDrone2019 dataset, LSCNet achieved 44.6% of mAP_{50}, 10.1% higher than the baseline model YOLOv10n, and surpassed the latter when the parameter amount was less than 10% (1.48 million) of YOLOv10m. Similarly, on the UAVDT dataset, LSCNet shows significant robustness in complex scenes such as night scenes and high-speed movements, and mAP_{50} is 5.06% higher than the baseline model.

Unlike traditional object detectors that rely on deep stacked convolution layers, our work proves that for small object detection based on UAVs, a simplified architecture focusing on shallow high-resolution feature processing can significantly improve efficiency and accuracy. LSCNet provides a feasible solution for deploying high-performance detection algorithms on resource-constrained UAV platforms.

7.2 Future Work

Although our method has shown good results, we also realize that relying on high-resolution feature maps to retain small object details will bring additional computing costs. In the future, the complexity of the model can be reduced by means of feature

distillation and network pruning, so as to reduce this trade-off and achieve more lightweight deployment. Future work will mainly focus on the following directions:

First, we will further optimize LSCNet to reduce its computational cost. Although LSCNet itself is already lightweight, processing high-resolution shallow features still imposes significant memory overhead. We plan to explore structured pruning technology and knowledge distillation to further compress the model. By refining knowledge from a larger, high-precision teacher model into a more compact student model, we aim to maintain the accuracy of small object detection while significantly reducing the number of floating-point operations (FLOPs).

Secondly, we plan to extend the efficiency analysis of the deep pyramid layers initially carried out on YOLOv10 to a wider range of pyramid networks and detection scenarios. We aim to study whether redundancy in deep semantic layers is a common phenomenon in small object detection tasks in different architectures. This analysis will guide the design of a more adaptive feature pyramid, which can dynamically allocate computing resources according to the scale distribution of the object.

References

[1] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.

[2] Y. Han, C. Wang, H. Luo, H. Wang, Z. Chen, Y. Xia, and L. Yun. Lrds-yolo enhances small object detection in uav aerial images with a lightweight and efficient design. *Scientific Reports*, 15(1):22627, jul 2025.

[3] Sabrina Jiang. A simple neural network, 2021.

[4] Jesudara Omidokun, Darlington Egeonu, Bochen Jia, and Liang Yang. Leveraging digital perceptual technologies for remote perception and analysis of human biomechanical processes: A contactless approach for workload and joint force assessment, 2024.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[7] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[11] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024.

[12] Ranjan Sapkota, Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, et al. Yolo advances to its genesis: a decadal and comprehensive review of the you only look once (yolo) series. *Artificial Intelligence Review*, 58(9):274, 2025.

[13] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024.

[14] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance

evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172:114602, 2021.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[16] Chen Xue, Yuelong Xia, Mingjie Wu, Zaiqing Chen, Feiyan Cheng, and Lijun Yun. El-yolo: An efficient and lightweight low-altitude aerial objects detector for onboard applications. *Expert Systems with Applications*, 256:124848, 2024.

[17] Yang Zhang, Xiaobing Chen, Su Sun, Hongfeng You, Yuanyuan Wang, Jianchu Lin, and Jiacheng Wang. Vehicle detection in drone aerial views based on lightweight osd-yolov10. *Scientific Reports*, 15(1):25155, jul 2025.

[18] D. Du et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *2019 ICCVW*, pages 213–226, 2019.

[19] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016.

[25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[26] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[27] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 9197–9206, 2019.

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[29] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[30] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[31] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 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*, pages 7464–7475, 2023.

[32] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. 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*, pages 2874–2883, 2016.

[33] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018.

[34] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1222–1230, 2017.

[35] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sodmtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 206–221, 2018.

[36] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[37] Huaxiang Zhang, Kai Liu, Zhongxue Gan, and Guo-Niu Zhu. Uav-detr: efficient end-to-end object detection for unmanned aerial vehicle imagery. *arXiv preprint arXiv:2501.01855*, 2025.

[38] Yuelei Wang, Ting Zhang, Liangjin Zhao, Lin Hu, Zhechao Wang, Ziqing Niu, Peirui Cheng, Kaiqiang Chen, Xuan Zeng, Zhirui Wang, et al. Ringmo-lite:

A remote sensing lightweight network with cnn-transformer hybrid framework. *IEEE transactions on geoscience and remote sensing*, 62:1–20, 2024.

[39] Dong Chen, Duoqian Miao, and Xuerong Zhao. Hyneter: Hybrid network transformer for multiple computer vision tasks. *IEEE transactions on industrial informatics*, 20(6):8773–8785, 2024.

[40] Qibin He, Xian Sun, Zhiyuan Yan, Bing Wang, Zicong Zhu, Wenhui Diao, and Michael Ying Yang. Ast: Adaptive self-supervised transformer for optical remote sensing representation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 200:41–54, 2023.

[41] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[42] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020.

[43] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2021.

[44] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

[45] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

[46] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4484–4496, 2025.

[47] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 13029–13038, 2021.

[48] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[49] Guangyi Tang, Jianjun Ni, Yonghao Zhao, Yang Gu, and Weidong Cao. A survey of object detection for uavs based on deep learning. *Remote Sensing*, 16(1):149, 2023.

[50] Wei Lu, Si-Bao Chen, Hui-Dong Li, Qing-Ling Shu, Chris H. Q. Ding, Jin Tang, and Bin Luo. Legnet: A lightweight edge-gaussian network for low-quality remote sensing image object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2844–2853, October 2025.

[51] Yuning Cui, Wenqi Ren, and Alois Knoll. Omni-kernel network for image restoration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(2):1426–1434, Mar. 2024.

[52] D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[53] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from

deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[54] Jingyang Wang, Jiayao Gao, and Bo Zhang. A small object detection model in aerial images based on cpdd-yolov8. *Scientific Reports*, 15(1):770, 01 2025.

[55] Minze Li, Yuling Chen, Tao Zhang, and Wu Huang. Ta-yolo: a lightweight small object detection model based on multi-dimensional trans-attention module for remote sensing images. *Complex & Intelligent Systems*, 10(4):5459–5473, 08 2024.

[56] Hezheng Wang, Jiahui Liu, Jian Zhao, Jianzhong Zhang, and Dong Zhao. Precision and speed: Lsod-yolo for lightweight small object detection. *Expert Systems with Applications*, 269:126440, 2025.